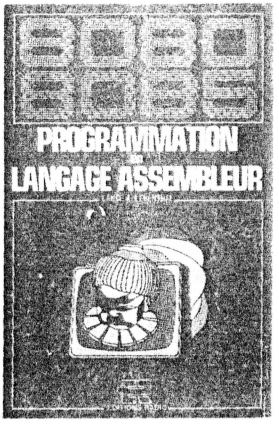
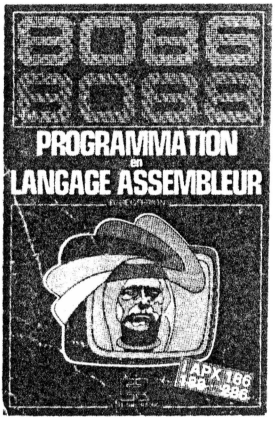
H. LILEN

TECHNIQUE-NORMES-APPLICATIONS DES INTERFACES POUR MICRO-ORDINATEURS

ÉDITIONS RADIO





ÉDITIONS RADIO

H. LILEN

Principes et applications des INTERFACES POUR MICRO-ORDINATEURS

— TECHNIQUES

— NORMES

— CONNEXIONS DES PERIPHERIQUES

Editions Radio
9, RUE JACOB - 75006 PARIS
TEL. 43.29.63.70

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.

© SECF Éditions Radio, Paris
1986

Tous droits de traduction, de reproduction et d'adaptation réservés pour tous pays.

Imprimé en France par Berger-Levrault, Nancy

Dépôt légal : juin 1986
Éditeur n° 1028 - Imprimeur : 778431
I.S.B.N. 2 7091 0986 7

AVANT-PROPOS

En 1981 et sous le titre « Interfaces pour microprocesseurs et micro-ordinateurs », les Editions Radio *publiaient un livre qui, en raison de son succès, devait être réédité en 1983. Mis à jour et complété, il passait de 272 pages à 350. Après divers « retirages » et en raison d'une demande toujours plus vive des lecteurs, une troisième édition était envisagée en 1985, elle aussi « complétée ».*

Hélas, le nombre de pages atteint eût été prohibitif pour un seul livre. Aussi a-t-il été décidé de le scinder très logiquement en deux ouvrages dont les thèmes respectifs sont :

- *Les interfaces matérielles, les normes, leurs applications, les interfaces réelles non normalisées, etc. C'est ce livre que vous avez entre les mains.*

- *Les télécommunications, incluant les modems, le minitel, les réseaux locaux et autres, les normes dans les communications, etc.*

Ce livre sur les interfaces regroupe un grand nombre d'informations qui, autrement, apparaissent éparses et difficiles à saisir. Pratique aussi, il montre par quoi se traduit l'application des interfaces, qu'elles répondent à des normes officielles ou de fait, ou à une absence totale de norme.

Vous y trouverez un grand nombre de schémas, tant synoptiques que d'applications pour de très nombreux types d'interfaces. En raison de la spectaculaire percée des « ordinateurs personnels » (PC) d'IBM, une large place a également été réservée à ses interfaces.

CHAPITRE I

AUTOUR DU MICROPROCESSEUR

Tout le problème des interfaces tourne autour de ce composant central qu'est le microprocesseur. On va examiner ici comment s'articulent les liaisons types dans un système de base, développé ensuite en micro-ordinateur.

Les mots-clés de ce chapitre

Microprocesseur : son organisation interne, 8
Circuit 8088 : brochage du microprocesseur, 9

Bus du microprocesseur, 10
Bus de systèmes à microprocesseur, 11
Bus d'extension de base : pour micro-ordinateurs, 14

1. ORGANISATION DE BASE D'UN SYSTÈME A MICROPROCESSEUR

Le microprocesseur exécute toutes les fonctions d'une *unité centrale de traitement* informatique. Ses quatre sections principales sont :

1. Une banque de registres, comprenant au moins *un accumulateur, un compteur ordinal, un registre d'instructions*, auxquels s'ajoutent des index, des pointeurs, etc.

2. Un décodeur d'instructions.

3. Une unité arithmétique et logique (ALU).

4. Des circuits de commande et de séquencement pilotés par une horloge à quartz.

MICROPROCESSEUR

La figure 1 représente la structure d'une telle *unité centrale de traitement*, UCT en abrégé, ou CPU, de l'américain « *central processing unit* ».

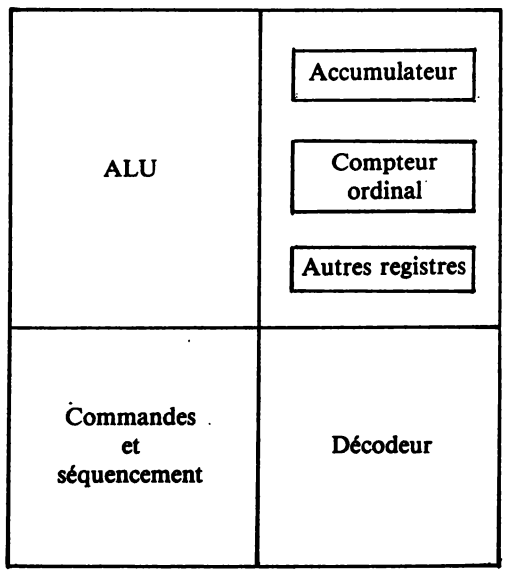
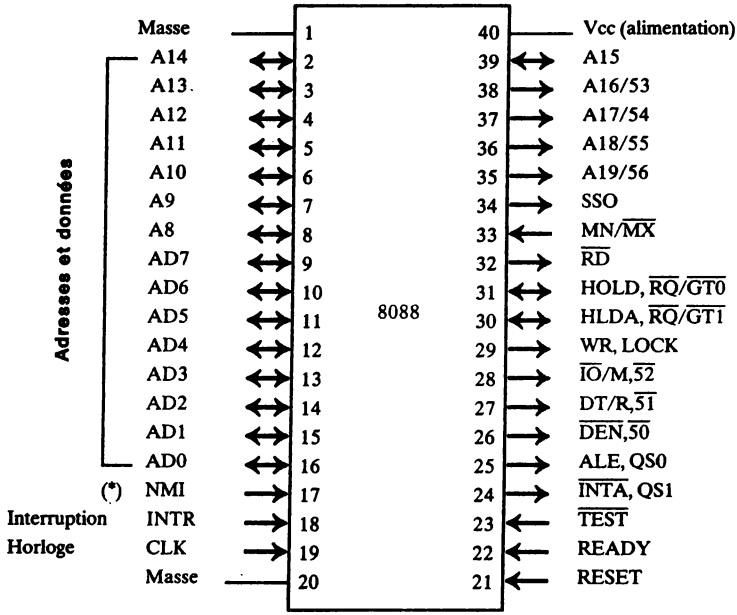


Fig. 1 — Organisation simplifiée d'un microprocesseur.

CIRCUIT 8088

Le microprocesseur communique avec les autres circuits et le monde extérieur via ses broches d'accès. Par exemple, le microprocesseur 8088 dispose de 40 broches d'accès ; elles sont indiquées dans la figure 2.



(*) Interruption non masquable

Fig. 2 — Brochage du microprocesseur 8088.

Ce 8088 émet des adresses sur 20 bits (lignes AØ à A19), ce qui lui permet d'adresser *un million* de cellules en mémoire (de 00000 à FFFFF en hexadécimal). Les données transitent sur 8 bits, via les lignes ADØ à AD7.

La différence avec le microprocesseur 8086 réside essentiellement dans ce bus de données : avec le 8086, les données transitent sur 16 bits, via les lignes ADØ à A15, appelée alors AD15.

Afin de simplifier la représentation de toutes ces lignes, on les regroupe en quatre grandes divisions :

- *les lignes d'adresses ;*
- *les lignes de données ;*
- *les lignes d'alimentation ;*
- les *lignes de commandes.*

Souvent, les lignes d'adresses et de données sont multiplexées (cas du 8088). Si l'on ne tient pas compte d'un éventuel multiplexage et si l'on néglige les lignes d'alimentation (implicites), la structure de principe devient celle de la figure 3.

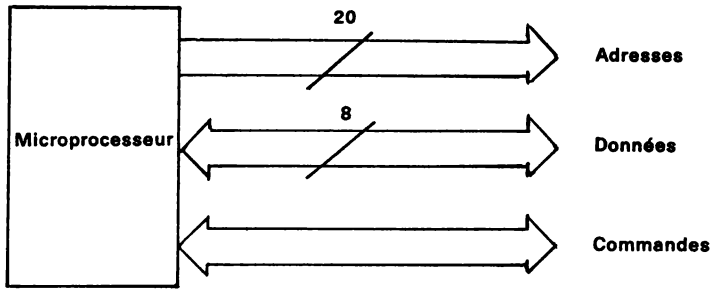


Fig. 3 — Organisation de principe des accès à un microprocesseur.

Afin de fonctionner, le microprocesseur doit encore être entouré de :

- *mémoires ;*
- *circuits d'interfaces* le mettant en relation avec le monde extérieur.

BUS DU MICROPROCESSEUR

Les circuits mémoires, tout comme les entrées-sorties (les E/S), peuvent être multipliés. Les lignes d'adresses, de données et de commandes vont tous les relier au microprocesseur ; elles se comportent à l'image des autoroutes, véhiculant des quantités d'informations et desservant de nombreuses localités, d'où la notion de bus. On traitera donc de (fig. 4) :

- *bus d'adresse*, unidirectionnel (du microprocesseur vers les autres circuits) ;
- *bus de données*, bidirectionnel ;
- *bus de commandes*, uni ou bidirectionnel, selon les commandes impliquées.

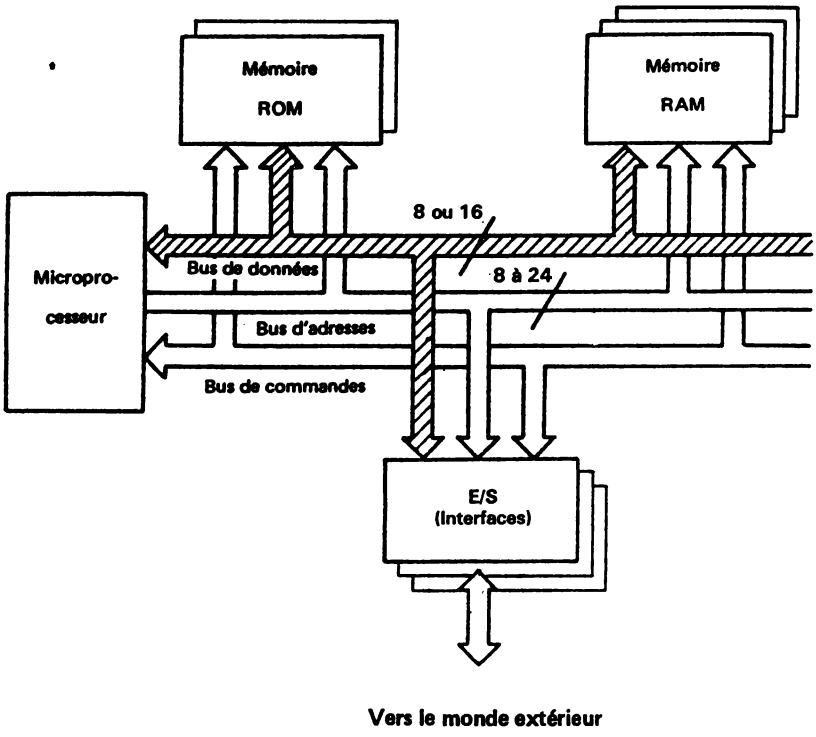


Fig. 4 — Les « bus » d'un système micro-ordinateur. Ce sont des voies en parallèle de liaison desservant de nombreux circuits. Ils peuvent être spécialisés, ce qui est le cas, ici.

2. LES DIVERSES ORGANISATIONS INTERNES

BUS DE SYSTEMES A MICROPROCESSEUR

L'organisation interne des bus peut revêtir de très multiples aspects. Ainsi, la figure 5 montre un bus unique desservant le système à microprocesseur, interne et externe. Le risque de saturation est grand si on lui connecte de nombreux éléments. On peut le scinder en deux, en distribuant ses fonctions entre mémoires et entrées-sorties (fig. 6) ou encore en prévoyant un bus interne au système ou « bus local », puis externe et permettant alors toutes extensions (fig. 7). Une structure triple pourra correspondre à bien des variantes, et par exemple à celle adoptée par *Intel* pour des mémoires à double accès (fig. 8) ; on trouve ici un bus interne, un bus d'accès à la RAM et enfin le bus externe, normalisé sous le nom de Multibus.

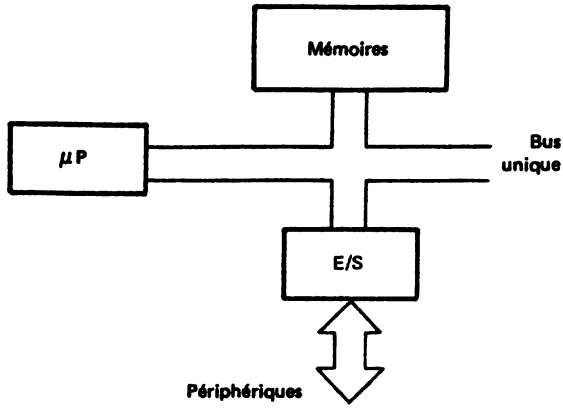


Fig. 5 — Système à bus unique.

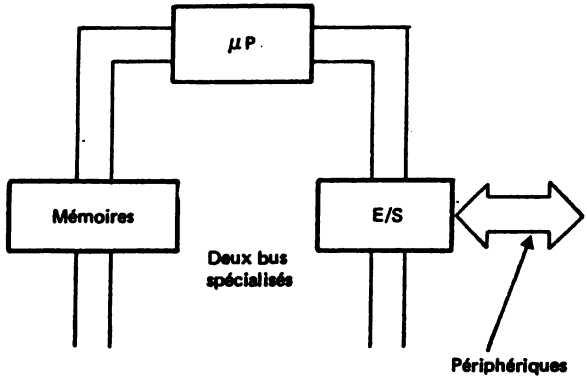


Fig. 6 — Double bus avec séparation des fonctions.

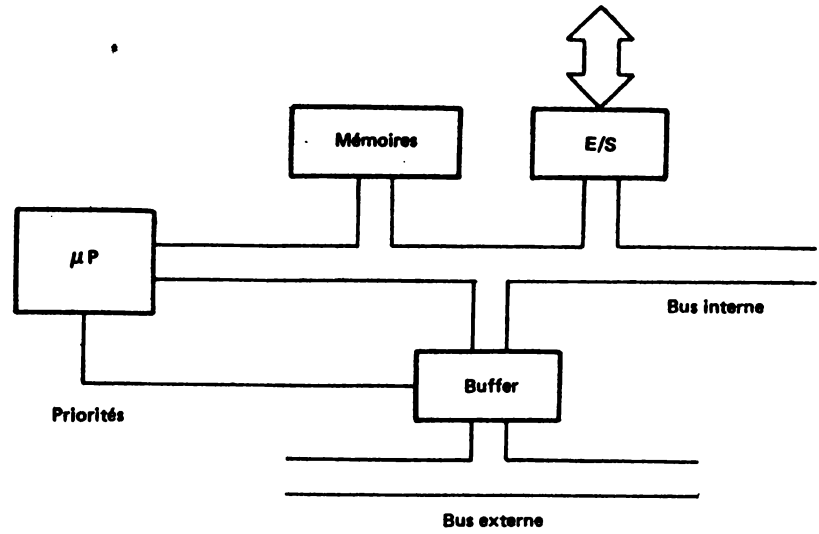


Fig. 7 — Double bus, interne et externe.

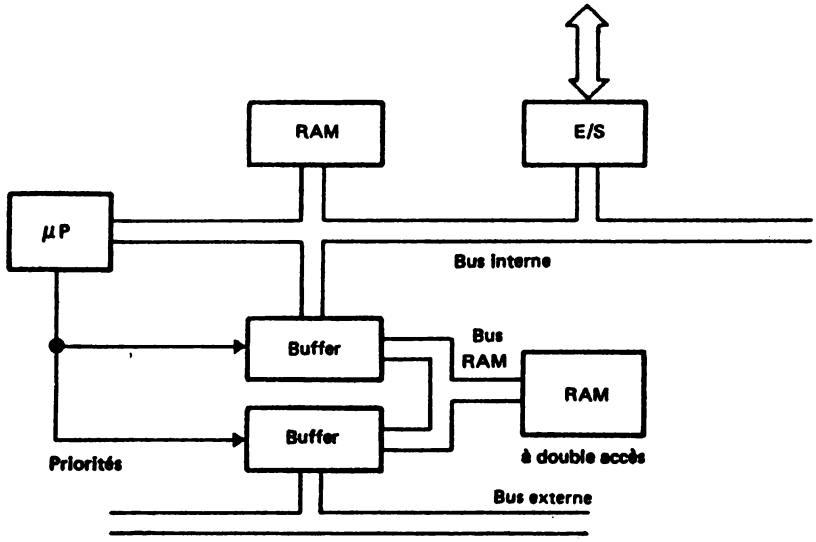


Fig. 8 — Triple bus pour mémoire à « double accès » : elle peut être atteinte soit par le système dessiné ici, soit et tout à fait indépendamment, par un autre système.

3. LE BUS D'EXTENSION DE BASE

L'extension du système, c'est-à-dire l'association au microprocesseur de nombreux circuits d'interfaces, de mémoires et de services annexes, se fait en les connectant à ces divers bus. La charge électrique qui en résulte pour le microprocesseur peut être importante, aussi intercale-t-on généralement des « circuits tampons » ou *buffers*. Ainsi, le schéma de principe complet devient-il celui de la figure 9.

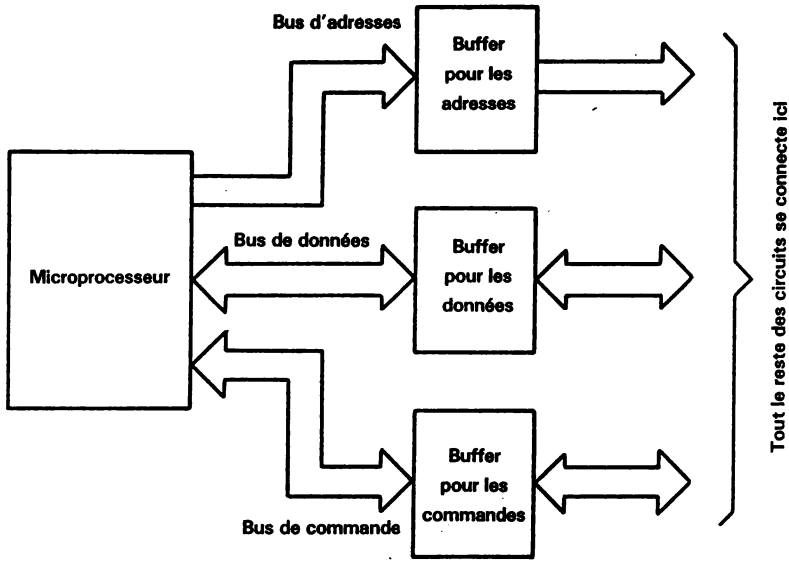


Fig. 9 — Les buffers sont des circuits tampons introduits sur les bus.

BUS D'EXTENSION DE BASE

A partir d'une telle structure, on peut concevoir un connecteur ou une série de connecteurs en parallèle alimentés par ces bus. Il suffira alors d'enficher dans ces connecteurs des cartes d'extension, mémoires, graphique-couleur, imprimante, etc. C'est ce qui est fait sur l'IBM-PC où les 5 connecteurs disponibles sont des éléments à 62 broches dans leur version complète, recevant :

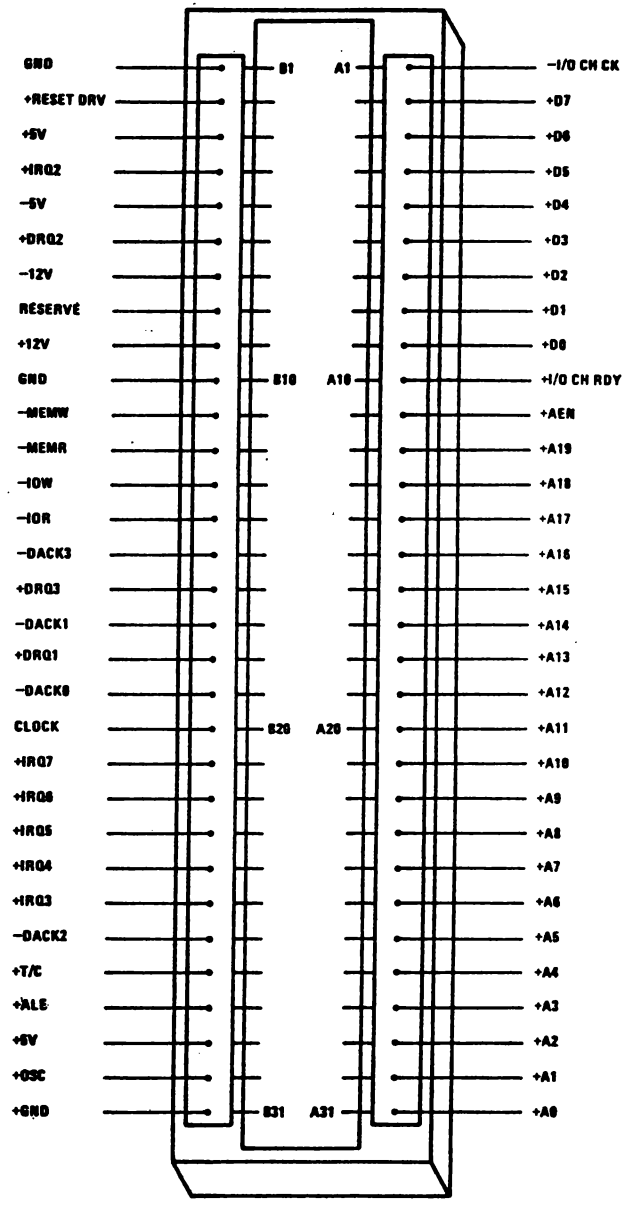


Fig. 10 — Le connecteur d'extension 62 broches de l'IBM-PC.

* les 20 lignes d'adresse ;
* les 8 lignes de données ;
* des lignes d'alimentation ;
* les lignes de commandes ;
* mais aussi quelques lignes supplémentaires.

En fait, ce bus d'extension est totalement démultiplexé (adresses et données sont distinctes) et se compose de (fig. 10) :

- 8 lignes de données, bidirectionnelles ;
- 20 lignes d'adresses ;
- 6 niveaux d'interruption ;
- 3 canaux d'accès direct en mémoire (DMA) ;
- 4 niveaux de tensions d'alimentation (+5 et −5 V, +12 et −12 V) et la masse ;
- des lignes de contrôle pour la mémoire ou les entrées-sorties, en lecture ou écriture, des lignes d'horloge ;
- des lignes commandant le rafraîchissement des mémoires dynamiques ;
- une ligne de test.

Nous reviendrons plus en détail sur ce bus dans le chapitre spécifique consacré aux bus d'extension des systèmes.

CHAPITRE II

LIAISONS EN PARALLÈLE

En numérique, un « mot » d'information comporte plusieurs bits. Les microprocesseurs traitent couramment des mots de 4 bits, 8, 16 ou 32 bits, par exemple. Ces bits sont représentés par un niveau de tension, niveau haut *pour le 1, niveau* bas *pour le zéro (en logique positive). Ils peuvent être transmis en* parallèle *ou en* série. *On va examiner, ici, ce premier mode.*

Les mots-clés de ce chapitre

Liaison parallèle : les principes, 18
Circuit 8255 : interface parallèle, 19
PIA : interface parallèle, 19
Imprimante en commande parallèle, 21

Centronics : norme pour commande d'imprimante parallèle, 21
Acknowledge : signal d'échange pour imprimante, 22
Busy : signal d'échange pour imprimante, 22
Strobe : signal d'échange pour imprimante, 22

Câble pour imprimante parallèle, 24
Connecteur 25 broches d'imprimante parallèle (IBM-PC), 25
Connecteur 36 broches d'imprimante parallèle (IBM-PC), 26
Clavier : commande par interface parallèle, 30

Clavier de l'IBM-PC : commandes et liaisons, 32
Feux de croisement : commande industrielle en mode parallèle, 35

1. PRINCIPES

LIAISON PARALLELE

Dans une transmission parallèle, il faut autant de fils ou de connexions qu'il y a de bits par mots. Les bits sont transmis en parallèle, mais les mots le sont en série, l'un après l'autre. La figure 1 symbolise une liaison parallèle sur 8 bits, où les huit connexions sont représentées.

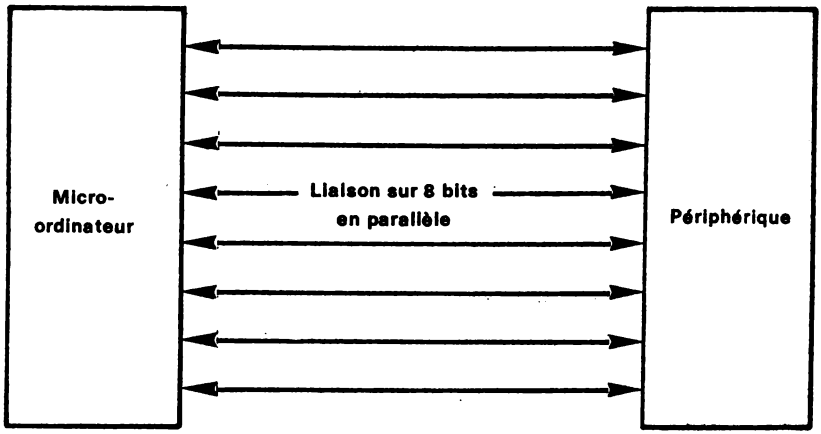


Fig. 1 — Exemple de liaison parallèle, sur 8 bits.

Nous conserverons ici l'exemple de liaisons parallèles sur 8 bits afin de traiter de *données* (et non d'adresses). Ce mode s'est, en effet, très largement développé, en particulier grâce au codage des caractères alphanumériques ; le code ASCII *(« American Standard Code for Information Interchange »)* fait appel à 7 ou 8 bits. La transmission sur un octet est, aussi, typique aussi bien des « anciens » microprocesseurs 8 bits (8080, 8085, 6800, Z80...) que du 8088.

Généralement, on passera par des circuits spécialisés d'interface, appelés *circuits d'entrées-sorties parallèles*. Ceux-ci peuvent souvent piloter plusieurs périphériques sur 8 bits et organiser un dialogue ou des échanges « dirigés » avec eux.

En outre, on leur demandera souvent d'être *programmables*, ce qui signifie qu'on voudra :

- transformer des lignes extérieures en *entrées* ou en *sorties*, à volonté ;
- gérer les échanges selon plusieurs méthodes possibles.

CIRCUIT 8255

L'un des circuits d'E/S programmables les plus courants est le 8255 de *Intel*. Il dispose de 4 groupes de 4 à 8 lignes dirigées vers le monde extérieur, selon le schéma de principe de la figure 2 ; la figure 3 donne son synoptique interne.

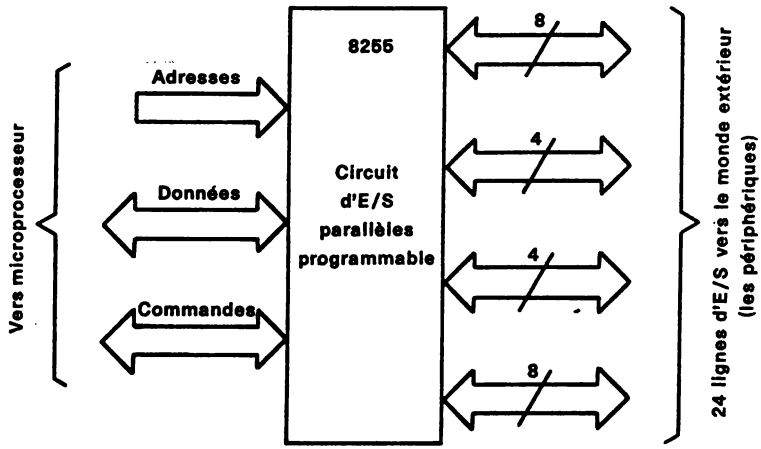


Fig. 2 — Principe de l'intervention du 8255, circuit d'entrées-sorties parallèles, programmable.

PIA

En symétrie et avec la famille *Motorola* des 6800, on trouve un circuit comparable et très connu, le « PIA » dont le montage avec un microprocesseur est donné dans la figure 4. Tout cela est toutefois très classique et est expliqué dans les manuels consacrés aux microprocesseurs, aussi n'est-il pas nécessaire de s'y attarder ici si ce n'est pour examiner les applications réelles.

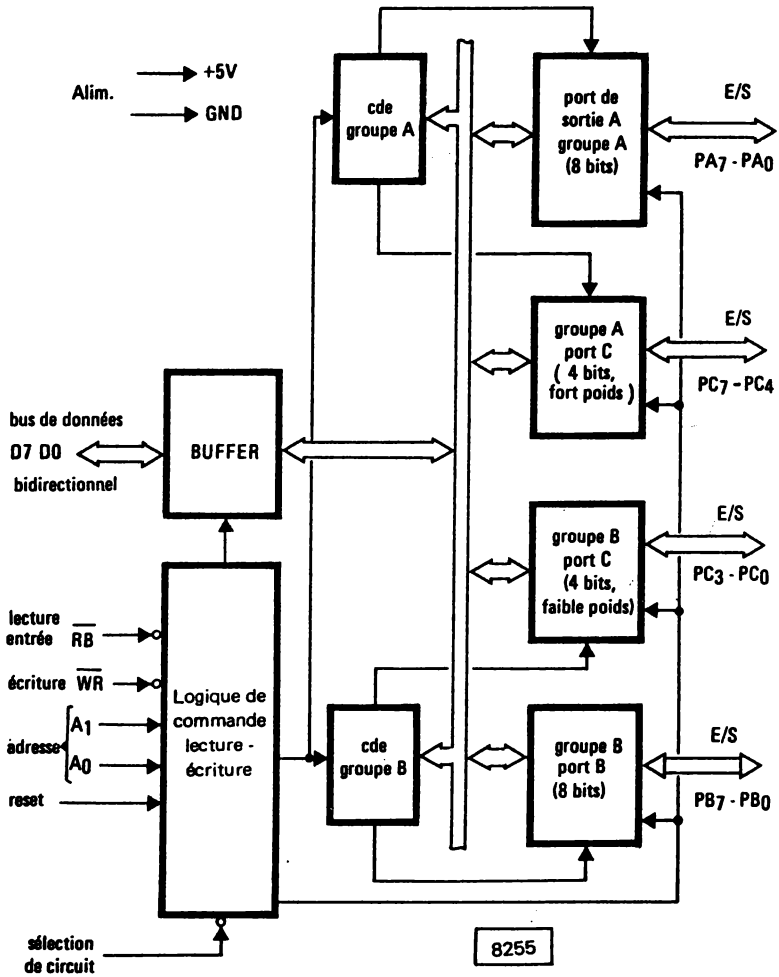


Fig. 3 — Constitution interne d'un circuit classique d'entrées-sorties en parallèle, le 8255. A gauche figurent les liaisons vers le microprocesseur, à droite celles allant vers le monde extérieur. C'est un circuit programmable : ses fils « extérieurs » peuvent être programmés en entrées ou en sorties.

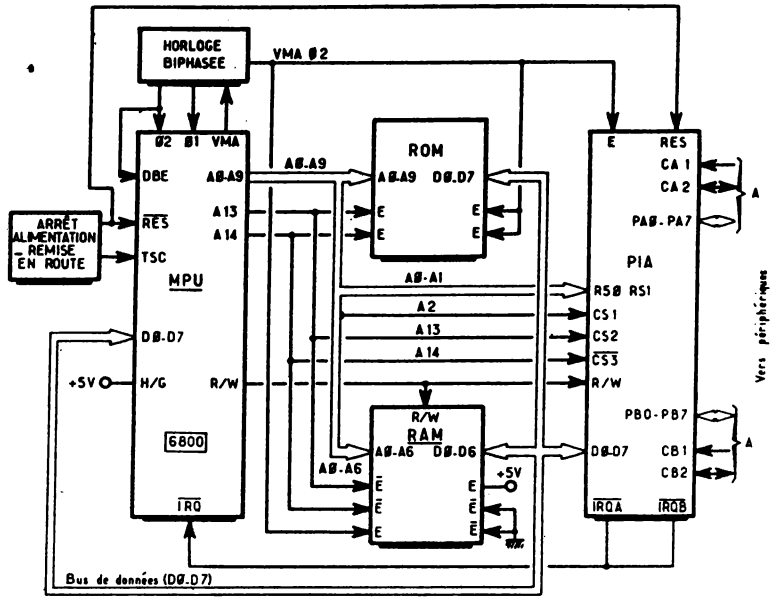


Fig. 4 — Montage type d'un circuit classique d'entrées-sorties parallèle dans un système à microprocesseur 6800. Le circuit d'E/S est le « PIA », initiales de « Peripheral Interface Adapter ». Ses deux « ports » d'E/S sur 8 bits sont programmables, en entrées ou en sorties.

2. COMMANDE D'UNE IMPRIMANTE : LA LIAISON CENTRONICS

IMPRIMANTE : COMMANDE PARALLELE
CENTRONICS

La commande d'une imprimante se fait en mode *série* ou en mode *parallèle*. En mode parallèle, le câble et les connecteurs sont plus complexes ; en outre, la liaison se fait sur des niveaux dits « TTL », ceux de la logique intégrée bipolaire dite « logique à transistors et transistors » qui sert de référence. Ces niveaux sont compris entre 0 et 5 volts. Le câble devra donc être court (deux à trois mètres, au maximum). C'est toutefois le mode de liaison le plus « naturel ».

Fabricant des imprimantes, la société *Centronics* a imaginé une liaison parallèle appelée de son nom et qui s'est imposée comme une norme de fait. C'est à elle qu'on va se référer.

Attention, toutefois : il ne s'agit nullement d'une norme rigoureuse. Elle connaît bien des variantes, en pratique, en particulier en câblage.

On va d'abord examiner son application à l'IBM-PC. Du côté du micro-ordinateur, une carte d'interface spécifique doit être insérée dans l'un de ses connecteurs internes ; c'est d'ailleurs une carte double desservant aussi le moniteur monochrome. L'imprimante est dotée de sa propre électronique de commande. Ces dispositifs sont suffisamment complexes (l'imprimante est dotée de son propre microprocesseur) pour qu'on les détaille ici. L'essentiel réside dans le principe de la liaison, qui est le suivant.

ACKNOWLEDGE
BUSY
STROBE

Les données sont transmises sur huit fils en parallèle. Lorsqu'elles sont émises par le micro-ordinateur et stables, celui-ci émet un ordre d'acquisition vers l'imprimante ; c'est le « *Strobe* », ou l'échantillonnage ; il constitue, en quelque sorte, une synchronisation. L'imprimante s'exécute et répond aussitôt par un signal « *Je suis occupée* », ou *Busy*. Dès qu'elle en a terminé, elle émet un signal témoignant que cette acquisition s'est faite *(« Acknowledge »)* et qu'elle est disponible pour la donnée suivante.

Par conséquent :

- **La synchronisation se fait avec *Strobe*.**
- **L'échange est géré via *Acknowledge* et/ou *Busy*.**

La vitesse d'échange est de 1 000 caractères par seconde au maximum. Son chronogramme est dessiné dans la figure 5.

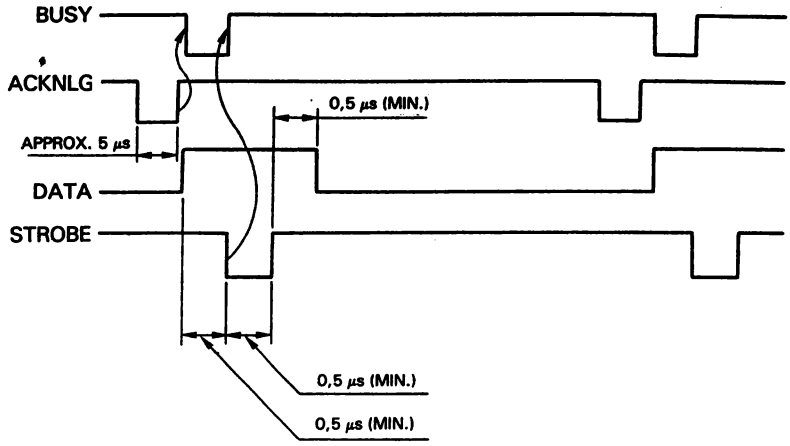


Fig. 5 — Principe de l'échange « en poignée de main » entre le micro-ordinateur et l'imprimante.

Les règles du dialogue peuvent être illustrées par la figure 6, indiquant le sens des échanges.

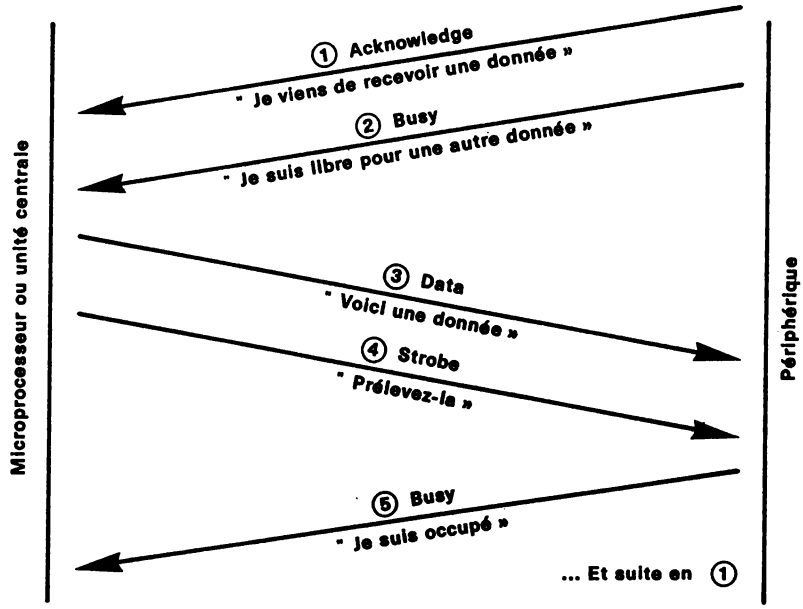


Fig. 6 — Le dialogue entre le microprocesseur et le périphérique.

Bien que les diverses électroniques de commande soient assez complexes et fassent appel à leurs propres microprocesseurs spécialisés, tant sur la carte d'interface du micro-ordinateur que dans l'imprimante, la commande des lignes reste très classique. On fera souvent appel à un étage buffer, une porte type 74 LS, ainsi que le montre la figure 7.

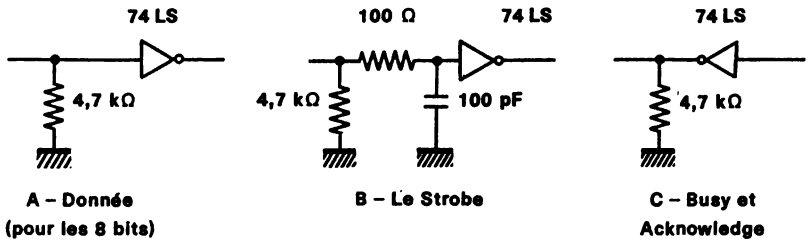


Fig. 7 — La commande de quelques lignes spécifiques : donnée et Strobe, vers l'imprimante, et Busy ou Acknowledge, de l'imprimante. Il s'agit là de montages types.

3. LE CÂBLE DE LIAISON A L'IMPRIMANTE

CABLE POUR IMPRIMANTE PARALLELE

La liaison entre le micro-ordinateur et l'imprimante est assurée à l'aide d'un câble spécifique. Nous allons nous référer au câble de l'IBM-PC pour l'imprimante graphique *IBM*, ou *Epson*. Si l'on reprend la notation de la logique *positive* pour laquelle le niveau *haut* (niveau 1) est le +5 V et le niveau *bas* (le zéro), le 0 V, on peut dresser la liste des fils de ce câble et leurs fonctions de la façon suivante :

• *8 fils pour la donnée*, sur 8 bits, en niveaux TTL ; la durée d'un niveau doit être de 2 µs au moins.

• *Un fil de synchronisation*, appelé *Strobe*. Normalement, à l'état *haut*, il devient actif en passant à l'état *bas* pendant au moins 0,5 µs et déclenche alors la lecture de la donnée par l'imprimante.

• *Un fil d'acquisition par l'imprimante (Acknowledge)* : une impulsion négative de 0,5 µs témoigne que l'imprimante a bien reçu la donnée et qu'elle est disponible pour la suivante.

• *Un fil d'occupation (Busy)* : au niveau *haut*, il indique que l'imprimante n'est pas en mesure de recevoir des données car : 1) elle est déjà en train de le faire ; 2) elle est en cours d'impression ; 3) elle est désélectionnée *(off-line)* ; 4) une erreur d'état s'est manifestée.

• *Un fil pour absence de papier (Out of paper*, ou PE, pour *Paper End)* : il passe au niveau haut lorsqu'il n'y a plus de papier.

• *Un fil de sélection* (*Select* ou SLCT) : au niveau haut, il indique que l'imprimante a été sélectionnée.

• *Un fil à auto-alimentation du papier (Auto-feed XT)* : lorsqu'il est au niveau *bas*, ce signal provoque un saut de ligne automatique après impression.

• *Un fil d'initialisation (Initialize printer)* : c'est un *reset* pour l'imprimante lorsqu'il passe à l'état *bas* pendant au moins 50 µs. Le buffer de l'imprimante est vidé.

• *Un fil d'erreur (Error)*, qui passe au niveau *bas* lorsque : 1) il n'y a plus de papier ; 2) l'imprimante est désélectionnée *(off-line)* ; 3) un état d'erreur a été détecté.

• *Un fil de sélection d'entrée (Select imput* ou *SLCT IN)*, qui autorise l'imprimante à accepter des données lorsqu'il est à l'état bas.

CONNECTEUR 25 BROCHES D'IMPRIMANTE PARALLELE (IBM-PC)

Il s'y ajoute les retours de masses *(ground)* pour tous ces fils, ce qui tend à constituer une situation plus complexe que ne le prévoit généralement celui qui veut réaliser soi-même son câble. Du côté *interface imprimante parallèle* du micro-ordinateur (toutes les broches sont utilisées en principe), les brochages sont indiqués figure 8 :

CONNECTEUR 25 BROCHES IBM-PC POUR IMPRIMANTE PARALLÈLE

Broche	Fonction	De (←) ou vers (→) l'imprimante	Niveau actif
1	Strobe	→	Bas
2 à 9	Donnée sur 1 octet	→	—
10	Acknowledge	←	Bas
11	Busy	←	Haut
12	Out of paper	←	Haut
13	Select	←	Haut
14	Auto feed	→	Haut
15	Error	←	Haut
16	Initialize printer	→	Haut
17	Select input	→	Bas
18 à 25	Ground	—	—

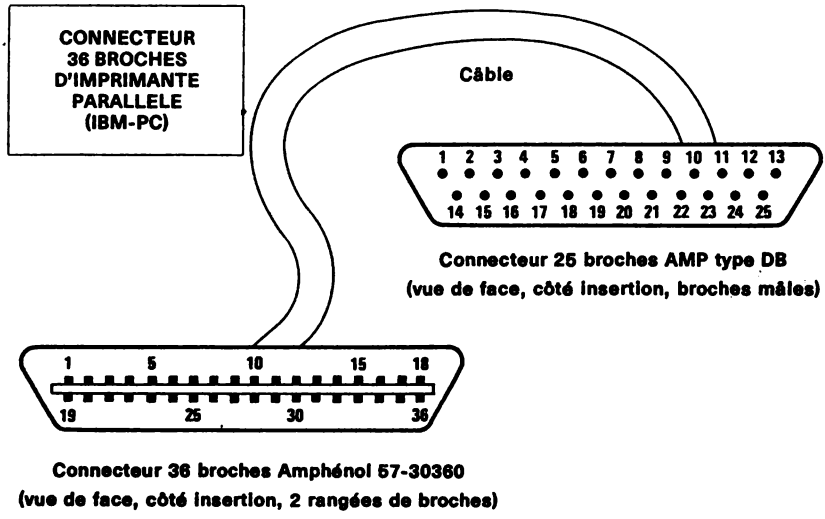

CONNECTEUR 36 BROCHES D'IMPRIMANTE PARALLELE (IBM-PC)
Câble
1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25
Connecteur 25 broches AMP type DB
(vue de face, côté insertion, broches mâles)
1 5 10 15 18
19 25 30 36
Connecteur 36 broches Amphénol 57-30360
(vue de face, côté insertion, 2 rangées de broches)

Fig. 8 — Allure des connecteurs du câble de liaison parallèle à l'imprimante (IBM-PC). Les connecteurs sont vus côté insertion, de face.

Méfiez-vous des indications de *niveaux actifs*, parfois définis arbitrairement (quel est, vraiment, le niveau actif ?). Sur l'imprimante IBM graphique matricielle (ou l'Epson...), on trouve un connecteur 36 broches *Amphénol* référence 57-30360 dont toutes les broches ne sont pas utilisées. En voici les attributions :

CONNECTEUR AMPHENOL 36 BROCHES
ENTRÉE PARALLÈLE DE L'IMPRIMANTE

Broche	Retour (masse)	Fonction	De (←) ou vers (→) l'imprimante	Actif au niveau
1	19	Stroble	→	Bas
2 à 9	20 à 27	Donnée	→	—
10	28	ACKNLG	←	Bas
11	29	Busy	←	Haut
12	30	Paper End	←	Haut
13	—	Select	←	Haut
14	—	Auto feed XT	→	Haut
15	—	Non connecté	Inutilisé	
16 17	— —	Zéro volt (masse) Masse du châssis		
18	—	Non connecté	Inutilisé	
19 à 30	—	GND (masses)		—
31	—	Init	→	Haut
32	—	Error	←	Haut
33	—	GND	Comme 19 à 30	—
34	—	Non connecté	Inutilisé	—
35	—	Rappelée en interne au niveau haut		—
36	—	Select in	→	Bas

Les retours de masses se font toujours via le blindage des fils. Certains d'entre eux peuvent se voir imposer un état, *bas* ou *haut*, grâce aux inverseurs situés sur l'imprimante.

Le câble reliant le micro-ordinateur à l'imprimante est câblé selon le modèle de la figure 9. Toutefois, on notera que :

• la ligne 14 *Auto feed* n'est pas indispensable et peut rester inutilisée ;

• dans certains cas, la ligne 18, côté imprimante, recevra le +5 V et ne devra pas être connectée ;

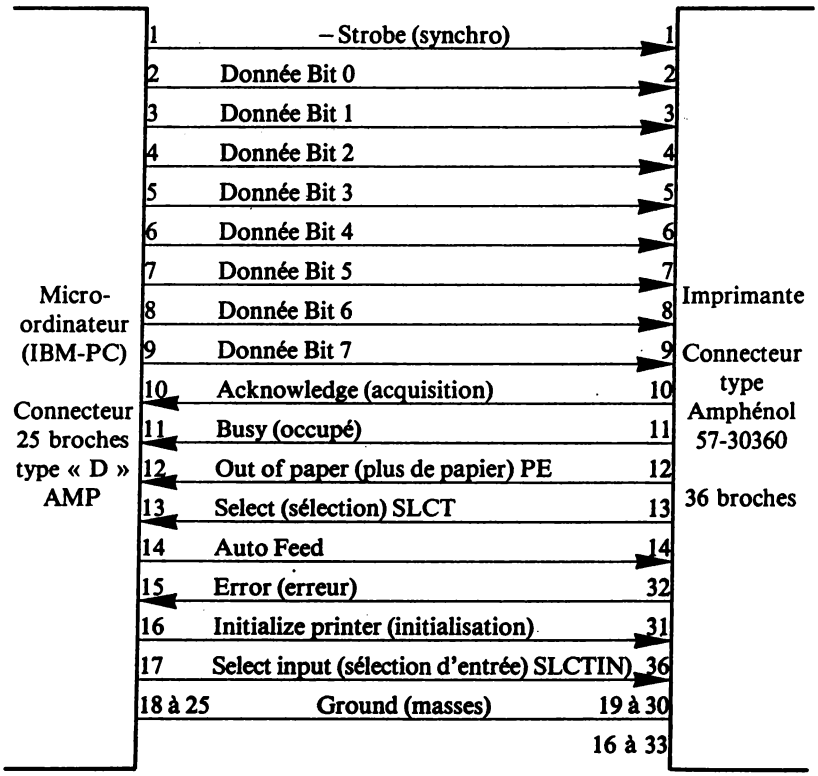

Micro-ordinateur (IBM-PC)
Connecteur 25 broches type « D » AMP
1 – Strobe (synchro) 1
2 Donnée Bit 0 2
3 Donnée Bit 1 3
4 Donnée Bit 2 4
5 Donnée Bit 3 5
6 Donnée Bit 4 6
7 Donnée Bit 5 7
8 Donnée Bit 6 8
9 Donnée Bit 7 9
10 Acknowledge (acquisition) 10
11 Busy (occupé) 11
12 Out of paper (plus de papier) PE 12
13 Select (sélection) SLCT 13
14 Auto Feed 14
15 Error (erreur) 32
16 Initialize printer (initialisation) 31
17 Select input (sélection d'entrée) SLCTIN 36
18 à 25 Ground (masses) 19 à 30
16 à 33
Imprimante
Connecteur type Amphénol 57-30360
36 broches

Fig. 9 — Liaison micro-ordinateur à imprimante en mode parallèle (connexion type Centronics pour l'IBM-PC).

• si les masses sont réunies au départ et à l'arrivée, il suffira d'un seul fil allant, par exemple, du 18 (connecteur 25 broches) au 19 (connecteur 36 broches).

De ce fait, un câblage réel type de câble pourra être celui de la figure 10 pour l'IBM-PC et l'imprimante graphique IBM ; il comportera 16 fils en tout et pour tout.

Certaines imprimantes en utiliseront encore moins. Ainsi, la Gemini de *Star* n'emploie que les 13 fils de la figure 11.

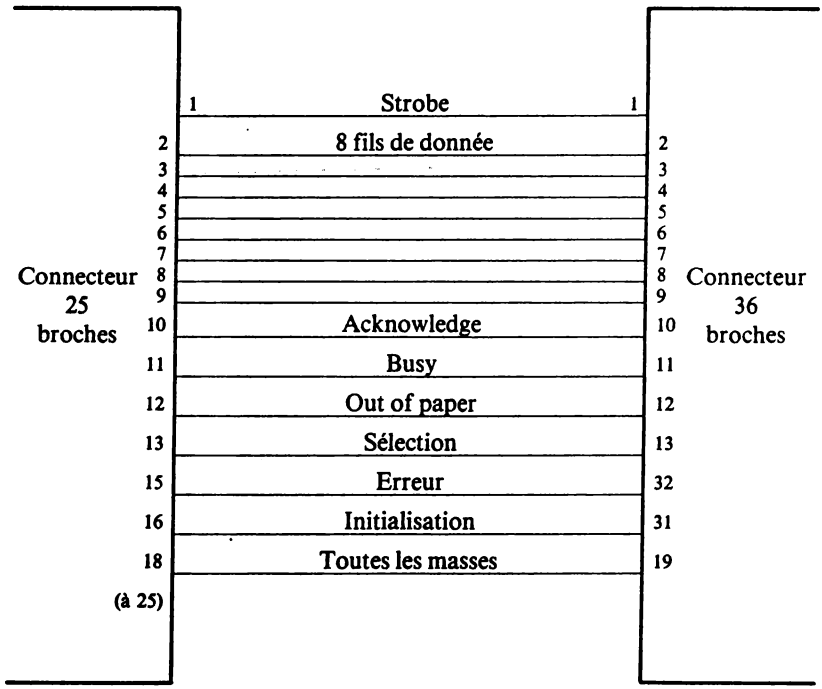


Fig. 10 — Câble réel pratique reliant l'IBM-PC à une imprimante graphique IBM, ou Epson, ou compatible.

Certaines imprimantes en utiliseront encore moins. Ainsi, la Gemini de *Star* n'emploie que les 13 fils de la figure 11.

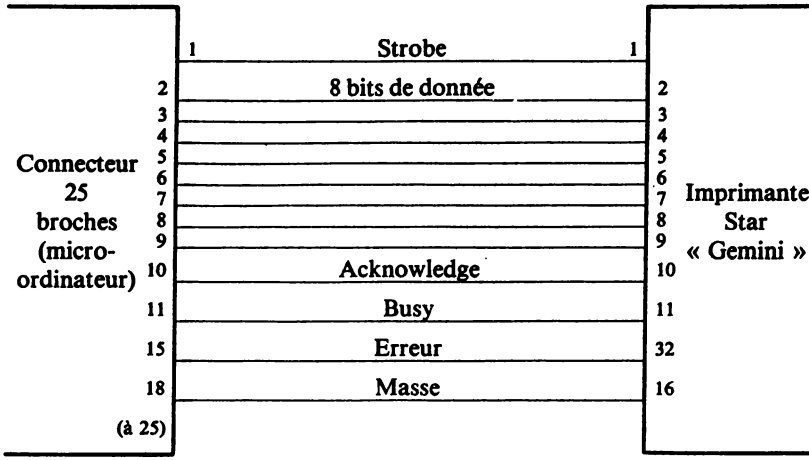


Fig. 11 — Certaines imprimantes réduisent le nombre de fils à 13.

4. COMMANDE DE CLAVIER

La commande d'un clavier relève, elle aussi, d'un mode de liaison parallèle mais, pour réduire le nombre de fils de liaison, on procède par un « balayage » des touches. Plusieurs méthodes sont employées ; voici le principe de l'une d'elles, appliqué à un clavier 16 touches, mais que l'on pourrait développer.

CLAVIER

Le mot étant supposé sur 8 bits, on le scinde en deux quartets dont l'un constituera une sortie, 0 ou 1, et l'autre une entrée. Le clavier à 16 touches est organisé en 4 colonnes et 4 rangées. Les colonnes sont attaquées par le quartet qui sort, les rangées par le quartet allant vers le microprocesseur (fig. 12) : *il n'y a aucun contact entre les colonnes et les rangées, au repos ; c'est la touche qui provoque le contact.* Le processus est le suivant :

— Au départ, le quartet sortant est à 1, les bits 4 à 7 restant des zéros (des résistances de rappel au 0 fixent ce potentiel). Si une touche est frappée, la rangée correspondante passe à 1 ; le NAND à 4 entrées détecte cet état en basculant : sa sortie, qui était à 1, passe à 0 et déclenche une demande d'interruption.

— Dès lors, le microprocesseur va émettre vers le circuit d'entrées-sorties des mots binaires chargés de balayer les colonnes. Il fournira successivement des quartets de 0001, puis 0010, puis 0100, puis 1000, cette séquence s'interrompant aussitôt qu'une touche enfoncée a été détectée.

— En effet, à chaque attaque d'une colonne par un 1, le quartet qui entre dans le circuit d'entrées-sorties est lu, chaque bit l'un après l'autre (bit 4, puis bit 5, puis bit 6, et enfin bit 7), jusqu'à ce qu'un bit à 1 soit détecté. Sinon, on passe à la colonne suivante.

— Le processus s'arrête dès lors que la touche enfoncée a été localisée par sa colonne et sa rangée.

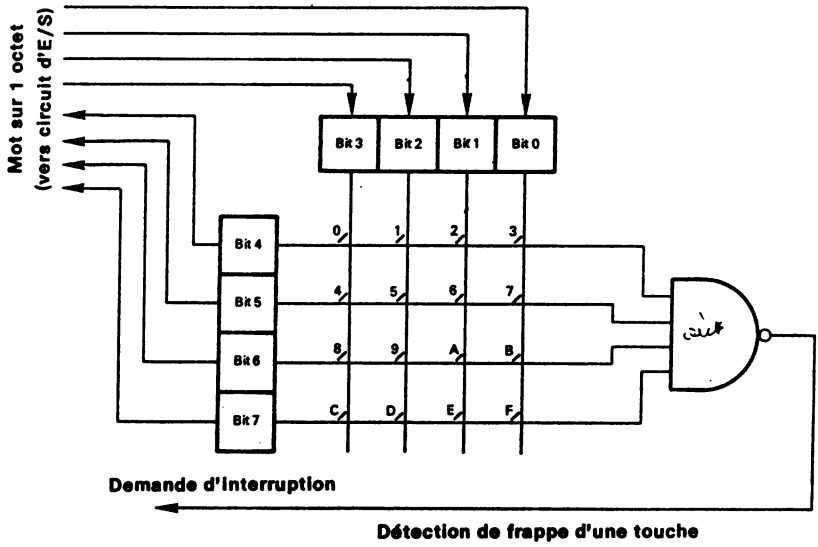


Fig. 12 — Exemple de commande d'un clavier 16 touches par un mot de un octet.

5. COMMANDE DU CLAVIER DE L'IBM-PC

CLAVIER DE L'IBM-PC

Les 83 touches du clavier de l'IBM-PC sont scrutées via deux ensembles électroniques :

- l'interface située sur la carte unité centrale ;
- l'électronique intégrée au clavier, comportant d'ailleurs elle-même son propre microprocesseur (un 8048).

De ce fait, la scrutation des touches se fait en mode parallèle, mais la liaison clavier à unité centrale est en mode série.

Le connecteur d'accès à l'unité centrale est un type « DIN » à 5 broches (fig. 13).

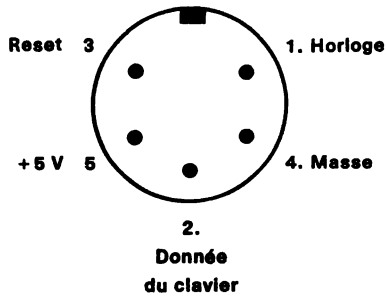


Fig. 13 — Le connecteur du clavier, à l'arrière de l'unité centrale.

Dans le clavier, l'électronique est pilotée par un microprocesseur 8048 disposant par lui-même de 24 lignes d'entrées-sorties à partir desquelles s'effectue la surveillance du clavier, à technologie capacitive. Son montage est esquissé dans la figure 14.

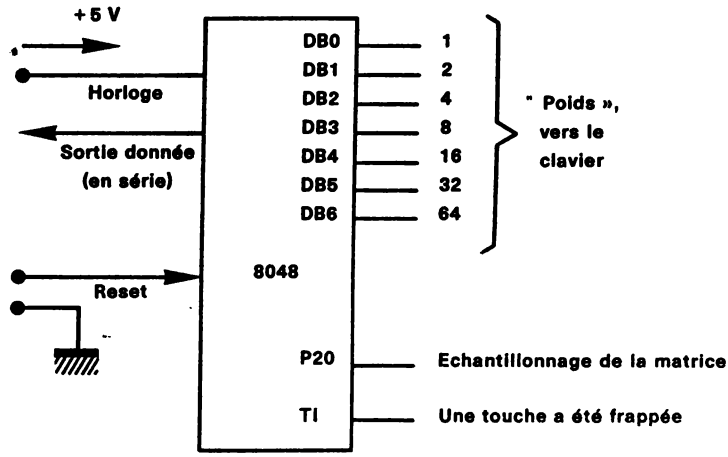


Fig. 14 — Un « micro-calculateur » 8048 gère le clavier en recueillant le « poids » des touches frappées.

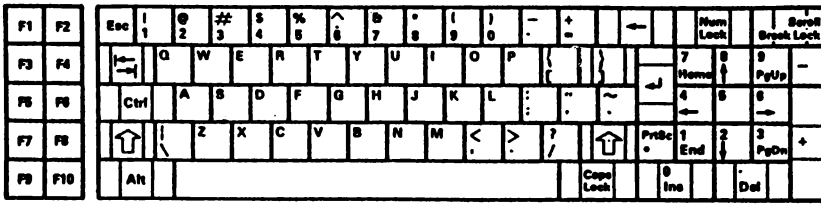
Les touches sont caractérisées par un *poids*, une valeur numérique, ce qui permet de leur affecter ensuite n'importe quel caractère ASCII. La figure 15 fournit ces valeurs.

Le poids des touches, de 1 à 83, est traduit en hexadécimal, de 01H à 53H. La plupart des touches génèrent un double code, au contact et à la rupture, distinguées par une différence de poids de 80H. Ainsi, la touche 01H au contact émet 80H à la rupture.

6. APPLICATIONS INDUSTRIELLES

Les accès parallèles se prêtent à de très nombreuses applications industrielles, chaque fil pouvant être connecté à un capteur, un actionneur, un contact, etc. Une très bonne illustration en est fournie par la commande de feux de croisement, dont la figure 16 illustre le principe.

Le clavier



Les valeurs affectées aux touches

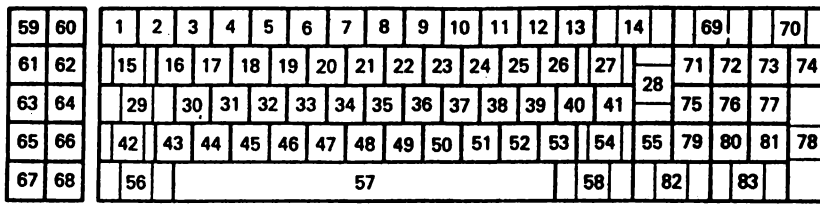


Fig. 15 — Valeurs affectées aux touches du clavier : leur poids, tel qu'il est utilisé lors de la scrutation.

FEUX DE CROISEMENT

Dans cette application, le mot de sortie, sur un octet, se voit attribuer la fonction de *mot d'états*. En effet, si l'on respecte la numérotation des fils du schéma, on constate que, pour mettre la voie B au vert et la voie A au rouge, il faut — en supposant qu'un 1 allume un feu et un 0 l'éteint — sortir le mot binaire 10000001. Le premier 1 (de plus faible poids) allume le vert de la voie B alors que celui de plus fort poids allume le rouge de la voie A.

Les bits du centre (3 et 4), inutilisés, restent à zéro ; ils pourraient servir aux feux piétons, par exemple. Ce mot, qui se coderait 81 en hexadécimal, n'a donc aucune signification autre que celle de l'état qu'il commande.

La séquence de commande de feux tricolores reviendrait, ici et en assembleur, à :

— loger le mot d'état dans l'accumulateur du microprocesseur ;

— sortir ce mot d'état ;

— temporiser pour la durée de cette situation ;

— loger le mot d'état suivant dans l'accumulateur (10000010, pour suivre notre exemple et avoir, cette fois, B à l'orange et A toujours au rouge) ;

— sortir ce mot, etc.

Il est évident que le circuit d'entrées-sorties ne peut commander directement des feux : il faudra introduire, en série, des amplificateurs, relais, ou même transistors VMOS de puissance, selon l'application.

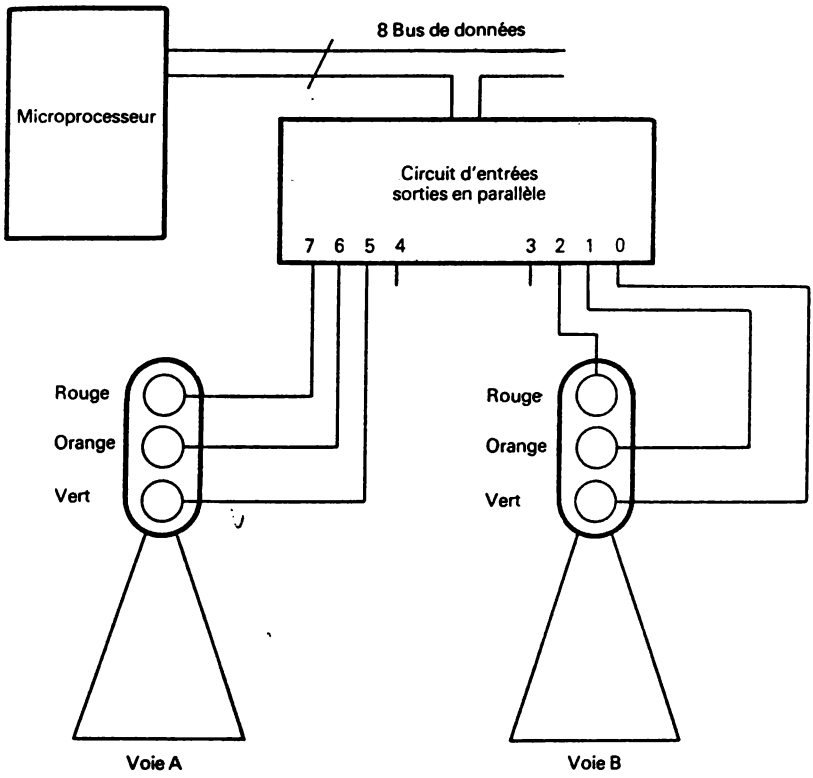


Fig. 16 — Principe d'une commande de feux de croisement avec des E/S parallèles sur 8 bits. Bien sûr, il faudra intercaler des amplificateurs ou des relais pour attaquer réellement les puissantes lampes des feux.

CHAPITRE III

LIAISONS SÉRIE ASYNCHRONES : PRINCIPES

Au mode de liaison parallèle s'oppose le mode série : toutes les informations sont transmises bit après bit sur une ligne unique. Ces liaisons série se décomposent à leur tour en modes asynchrone et synchrone. C'est aux liaisons série asynchrones qu'est consacré ce chapitre.

Les mots-clés de ce chapitre

Synchrone : mode, 38
Asynchrone : mode, 38
Start-stop : synonyme de mode asynchrone, 41
Baudot : code, 42

Code télex CCITT n° 2, 43
Codes CCITT, 43
ASCII : code, 45
Parité : pour correction d'erreurs, 46

Imparité : pour correction d'erreurs, 46
Conversion parallèle-série, 49
Conversion série-parallèle, 50
Détection du start : en mode asynchrone, 50

Faux départ : en mode asynchrone série, 50
Boucle 20 mA : liaison asynchrone série, 52
Boucle 60 mA : liaison asynchrone série, 53
Commande de téléimprimeur par boucle 20 mA, 54

Coupleur optique : pour liaisons série, 55
UART : circuit de liaison asynchrone, 56
ACIA : un circuit UART pour liaison asynchrone, 56

1. MODES SYNCHRONE ET ASYNCHRONE

MODES SYNCHRONE ET ASYNCHRONE

La liaison parallèle a pour défaut d'exiger beaucoup de lignes, ce qui se traduit par une belle consommation de fils de cuivre, d'accès aux circuits, de connecteurs, etc., ainsi d'ailleurs que pas mal d'électronique puisque, si l'on a besoin d'un buffer en cours de route, il faudra le multiplier par le nombre de lignes, par exemple. La liaison en série remédie à ces problèmes mais impose d'autres servitudes ; ainsi, au lieu de transmettre simultanément les huit bits d'un octet, comme le fait la liaison parallèle, elles les transmet l'un après l'autre sur la même voie. A vitesse égale, on comprend que la transmission d'un octet prenne huit fois plus de temps (en principe) ; en outre, il faudra généralement assurer une conversion parallèle à série à l'émission puisque le microprocesseur fonctionne, lui, en parallèle ; à l'inverse et à la réception, on procèdera à la conversion inverse, série à parallèle. De toutes façons, on n'évitera pas l'envoi supplémentaire d'informations *de dialogue*, soit sur la même ligne, soit sur des lignes distinctes (fig. 1).

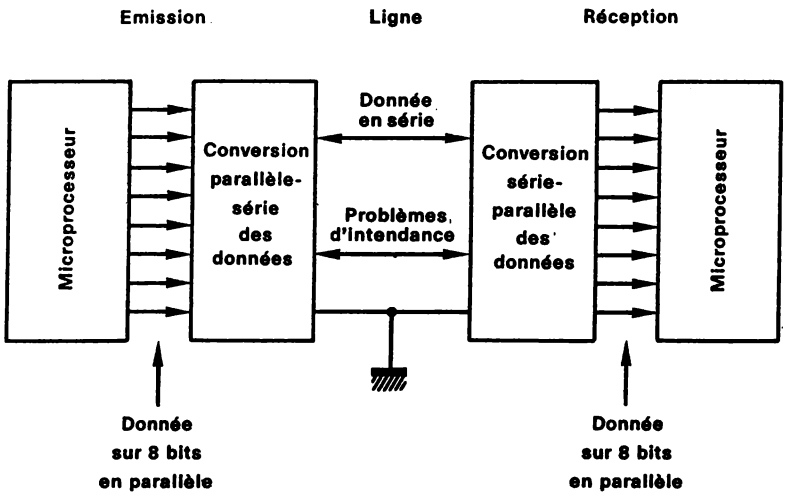


Fig. 1 — Principe d'une liaison série de microprocesseur à microprocesseur.

La liaison pourra, ou *devra* plutôt, être bidirectionnelle dans nombre de cas, ce qui implique l'usage de circuits d'interface fonctionnant dans les deux sens. Enfin, cette liaison série peut se concevoir :

• avec émission en une salve d'une grande masse de données, sans pause ni temps mort : c'est le *mode synchrone*, où une *horloge* cadence la transmission ;

• avec transmission des données au fur et à mesure qu'elles se présentent, avec des temps morts éventuels. C'est le cas typique d'une frappe au clavier, où l'émission des données suit la cadence de l'opérateur. C'est le mode *asynchrone*, étudié ici.

2. ORIGINE ÉLECTROMÉCANIQUE

Très employée, la *liaison série asynchrone* sera mieux comprise si l'on se réfère à son origine, la télégraphie électromécanique.

Rappelons au préalable que ce mode de liaison se fait *en série*, bit après bit, et en mode *asynchrone*, c'est-à-dire qu'un nouveau caractère (composé de plusieurs bits successifs) *peut être émis à tout moment*. Cela, à l'opposé du *mode synchrone* pour lequel l'instant de départ d'un nouveau caractère est rigoureusement situé dans le temps, en synchronisme avec une horloge. Pour l'instant, on considérera simplement que le mode asynchrone est moins astreignant mais moins rapide que le synchrone.

Soit donc le schéma de la figure 2, représentant un émetteur et un récepteur de télégraphie « start-stop », nom conservé qui reste synonyme de *série asynchrone*. Sur l'émetteur, lorsqu'on frappe un caractère au clavier qui lui est connecté : « Start » se ferme quel que soit le caractère, les contacts 1 à 5 se ferment en fonction d'un code. Au repos, ils sont tous ouverts.

Une piste externe regroupe des zones conductrices, isolées entre elles, repérées *Start, Stop* et par des numéros de 1 à 5. La zone *Stop* est reliée en permanence au *plus* de la batterie ; au repos, le bras mobile la court-circuite avec une piste intérieure C via des balais ; cette piste C sort sur la ligne, qui est donc *sous tension au repos*.

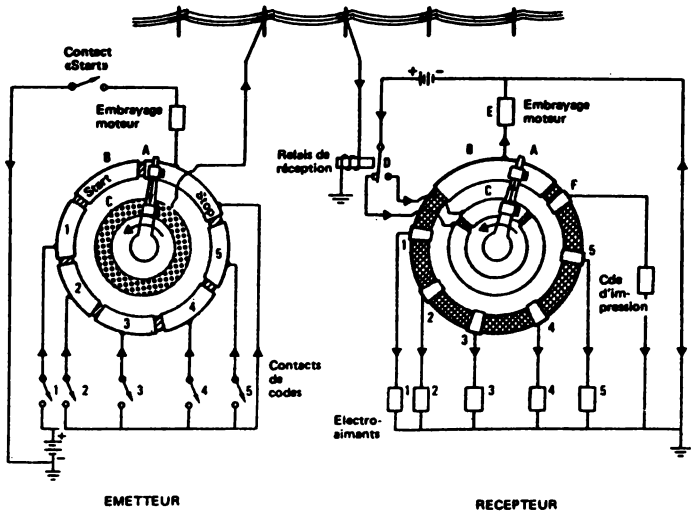


Fig. 2 — Principe d'un émetteur-récepteur télégraphique électromécanique. En passant à l'électronique, on conservera la structure des signaux transmis.

Si l'opérateur frappe une touche au clavier, le contact « Start » se ferme et commande un embrayage qui met en route un moteur à vitesse constante entraînant le bras mobile dans le sens du cercle trigonométrique. On va ainsi envoyer successivement sur la ligne des impulsions correspondant d'abord au niveau *bas* (zéro) en passant sur « Start », puis répondant à la combinaison du caractère frappé en passant sur les fractions de piste 1 à 5, jusqu'au retour à « Stop » où le bras s'arrêtera.

Du côté récepteur, on trouve un dispositif similaire où les contacts 1 à 5 sont remplacés par des bobines d'électro-aimants. Au repos, le courant qui passe par la ligne excite le relais D ; le bras mobile est à l'arrêt dans la position indiquée. Dès qu'une frappe a lieu à l'émetteur, son bras passe sur *Start*, ce qui interrompt le courant dans la ligne : le relais D du récepteur inverse sa position, ce qui excite l'embrayage du moteur du récepteur. Le bras mobile du récepteur va faire un tour de piste à la même vitesse que l'émetteur, mais, là, la piste est fractionnée en d'étroites zones conductrices qui vont « échantillonner » les impulsions émises, vers leur centre : selon leur niveau, les relais correspondant seront ou non excités.

Outre les cinq contacts de la piste du récepteur intervient un sixième qui, lui, détecte le retour sur *Stop* du bras de l'émetteur et commande alors la frappe

du caractère correspondant à la combinaison émise, et reçue par les électro-aimants.

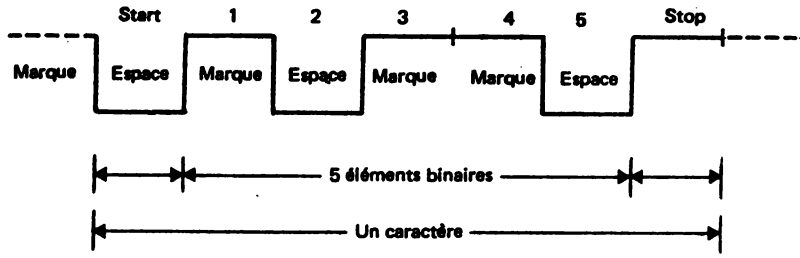


Fig. 3 — Exemple de signal série transmis (en code Baudot, à 5 moments). Il est précédé par un signal de « Start » (interruption du courant) et suivi par une impulsion et demie au niveau haut qui donne le « Stop » (indiquant la fin du caractère).

Tel est le principe de base des dispositifs télégraphiques électromécaniques, à codes à 5 bits. Par conséquent, le signal transmis par la ligne a la forme montrée par la figure 3. Au repos, un courant parcourt la ligne et témoigne de la disponibilité de l'équipement ; cet état « 1 » est appelé *marque*. Il est suivi par l'impulsion de *Start*, au niveau « 0 » encore appelé *espace*, puis par les 5 bits du code (c'est le code Baudot auquel on se réfère ici).

3. TRANSPOSITION ÉLECTRONIQUE : LE MODE « START-STOP »

START-STOP

L'électronique a repris ces mêmes principes. Si l'on reste en *logique positive* où le 1 est le niveau *haut* et le 0 le niveau *bas*, la ligne de transmission supporte un niveau *haut* au repos (fig. 4). Une coupure pendant un temps unitaire correspond à l'indication de l'envoi d'un caractère : c'est le signal de *start*. Ce caractère comprend de 5 à 8 bits, selon le code utilisé. Il est suivi ensuite par un « bit de parité » (ou *d'imparité*) dont on va voir le rôle, et par un bit, ou 1,5 bit, ou deux bits marquant la fin du caractère *(bits de stop)*. Après quoi la ligne reste à l'état *haut*, dans l'attente d'un nouveau cycle pour le caractère suivant.

Ainsi donc, la transmission des caractères est *asynchrone*, mais la série des bits qui représentent l'envoi d'un caractère, une dizaine ou une douzaine de bits, doit respecter des temps précis et rigoureusement calibrés : chaque bit se voit imparti un temps donné, sur lequel le récepteur est synchronisé, faute de quoi rien d'exact ne pourrait être détecté à la réception. C'est le front descendant du bit de start qui déclenche, à la réception, l'horloge de lecture.

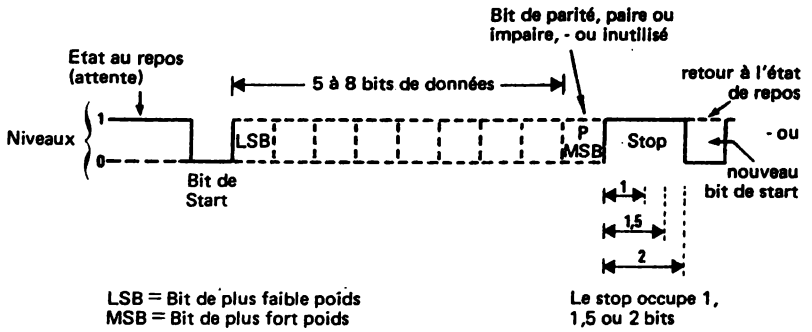


Fig. 4 — Transmission série asynchrone ; le signal comprend ici un bit de start, jusqu'à 8 bits de données, un éventuel bit de contrôle de parité, et un, un et demi ou deux bits de stop.

MODE ASYNCHRONE START-STOP

C'est donc parce que les caractères sont transmis en mode asynchrone et encadrés par des bits de *start* et de *stop*, que ce mode de transmission série a été encore appelé *mode start-stop*, ainsi qu'on l'a indiqué précédemment.

4. LES CODES

CODE BAUDOT

Un caractère est ainsi codé sur plusieurs bits. Le premier code à longueur constante, sur 5 bits, a été imaginé en 1874 par Emile Baudot et porte son nom ; il s'est manifesté sous de nombreuses variantes ultérieures, mais son défaut principal réside dans le faible nombre de combinaisons qu'il permet : 32 (sur 5 bits), ce qui ne suffit pas pour les seules lettres de l'alphabet et les chiffres : on utilise alors un artifice pour étendre ses possibilités.

CODE TELEX CCITT N° 2

Le télex utilise un code à cinq bits, normalisé sous le nom de CCITT n° 2, CCITT étant les initiales du *Comité Consultatif International des Téléphones et Télégraphes*. Ce code est représenté dans le tableau I : l'artifice pour étendre les possibilités de codage consiste dans l'utilisation des combinaisons 11111 et 11011 qui précisent respectivement si les caractères qui suivent sont *des lettres* ou *des chiffres* et *des symboles* (un artifice de ce type est utilisé pour les machines à écrire, avec la touche « majuscules »). Les caractères spéciaux sont : RC, pour *retour du chariot*, SP pour *espacement* (ou caractère *blanc*), et AL pour *avancement d'une ligne*.

TABLEAU I — LE CODE A 5 BITS NORMALISE CCITT N° 2

4 derniers bits		0000	0001	0010	0011	0100	0101	0110	0111
1er bit = 0	Mode « lettre »		T	RC	O	SP	H	N	M
	Mode « chiffre »		5		9			,	.
1er bit = 1	Mode « lettre »	E	Z	D	B	S	Y	F	X
	Mode « chiffre »	3	+	Prêt ?	?	'	6		/

4 derniers bits		1000	1001	1010	1011	1100	1101	1110	1111
1er bit = 0	Mode « lettre »	AL	L	R	G	I	P	C	V
	Mode « chiffre »		)	4		8	0	:	=
1er bit = 1	Mode « lettre »	A	W	J	Mode chiffre	U	Q	K	Mode lettre
	Mode « chiffre »	—	2	Son-nette		7	1	(	

NORMALISATION CCITT

La normalisation du *CCITT* est plus générale et est donnée dans le tableau II, les conditions A et Z étant à leur tour définies dans le tableau III selon les « recommandations » R44 et S12 du *CCITT*. La figure 5 illustre ce code avec une bande perforée télé-imprimeur à 5 canaux où l'on retrouve les codes 11111 et 11011 pour la sélection « lettres » et « chiffres ».

TABLEAU II — LE CODE CCITT N° 2 (VERSION GÉNÉRALE)

Caractères		Numéro du symbole	Code international n° 2 à 5 états (utilisé par le Télex)
Lettres	Chiffres		
A	—	1	ZZAAA
B	?	2	ZAAZZ
C	:	3	AZZZA
D	Qui êtes-vous ?	4	ZAAZA
E	3	5	ZAAAA
F	E	6	ZAZZA
G	%	7	AZAZZ
H		8	AAZAZ
I	8	9	AZZAA
J	Sonnette	10	ZZAZA
K	(	11	ZZZZA
L	)	12	AZAAZ
M	.	13	AAZZZ
N	,	14	AAZZA
O	9	15	AAAZZ
P	0	16	AZZAZ
Q	1	17	ZZZAZ
R	4	18	AZAZA
S	'	19	ZAZAA
T	5	20	AAAAZ
U	7	21	ZZZAA
V	=	22	AZZZZ
W	2	23	ZZAAZ
X	/	24	ZAZZZ
Y	6	25	ZAZAZ
Z	+	26	ZAAAZ
Retour chariot		27	AAAZA
Autre ligne		28	AZAAA
Lettres		29	ZZZZZ
Chiffres		30	ZZAZZ
Espace		31	AAZAA
Répétition		32	AAAAA

CODE ASCII

En fait, les codes se sont multipliés, surtout vers les années 60. L'un des plus utilisés est, actuellement, le code ASCII, abréviation de *American Standard Code for Information Interchange*, défini par l'*American National Standard Institution* (ou ANSI) et version US des normes CCITT n° 5. C'est un code à 7 bits, donné par le tableau IV ; dans le tableau V ont été définies les abréviations utilisées.

Le plus souvent, on utilise ce code en lui ajoutant un huitième bit, de *parité* ou d'*imparité*, afin de détecter d'éventuelles erreurs. C'est pourquoi on dit parfois qu'il s'agit d'un code à 8 bits.

TABLEAU III — LES CONDITIONS A ET Z DU TABLEAU PRÉCÉDENT SELON LES RECOMMANDATIONS R44 ET S12 DU CCITT

Condition	A	Z
Bits	0	1
Code start-stop	Espace (start)	Marque (stop)
Perforations (dans ruban papier)	Pas de trou	Trou présent
Liaison à tension unique	Pas de tension	+ V
Liaison à double tension	− V	+ V
Modulation d'amplitude	Non modulé	Modulé
Modulation de fréquence	Fréquence haute	Fréquence basse
Modulation de phase avec référence	Opposition de phase	En phase
Modulation de phase différentielle	Inversion	Pas d'inversion

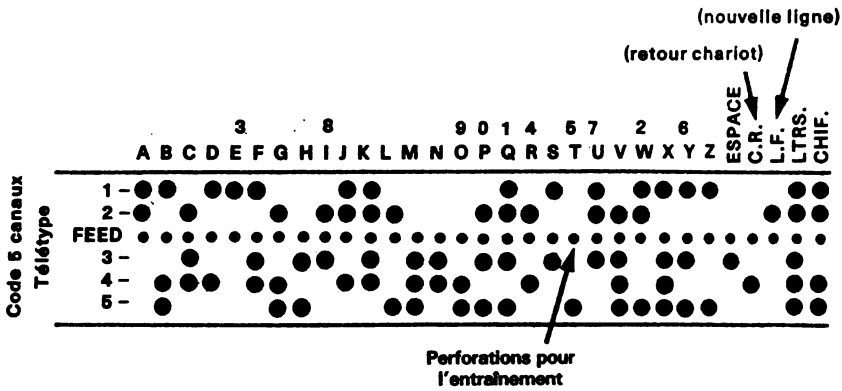


Fig. 5 — Exemple de bande perforée de télé-imprimeur à cinq canaux, avec détail du code.

Enfin, il peut être « complété » sur 8 bits afin que 256 caractères différents soient disponibles, ce qu'a fait *IBM* sur son PC, par exemple.

5. PROTECTION CONTRE LES ERREURS

Afin de déceler d'éventuelles erreurs intervenues au cours de la transmission, on utilise des *méthodes de détection d'erreurs* dont la plus simple est *le contrôle de parité.*

PARITE
IMPARITE

On dit d'un mot *binaire* qu'il est pair lorsque le total de 1 qu'il contient est pair. Par exemple, et sur sept bits, le mot 1010110 est pair, alors que que 0010110 est impair. Or, on peut se fixer la condition suivante : *à l'émission, on n'enverra que des mots pairs*. Par conséquent, on ne devra recevoir que des mots pairs également, faute de quoi c'est qu'une erreur (due à un parasite, à une perte d'information...) se sera produite au cours de la transmission. La convention inverse est également valable : on n'émet que des mots *impairs*, et par conséquent, on ne doit recevoir que des mots impairs ; le tout est d'arrêter l'une de ces deux formules, au choix.

Comment fait-on pour obtenir systématiquement des mots pairs, ou impairs ? Aux 7 bits du code ASCII, par exemple, on ajoutera un huitième bit *qui n'a d'autre signification que celle de fournir la parité souhaitée*. Ainsi, en « parité *paire* », le mot 1010100 deviendra 10101001 (on a ajouté un dernier 1), alors que le mot 0101101, déjà pair, ne recevra qu'un zéro supplémentaire pour devenir 01011010 *sur 8 bits*.

Si l'on recherchait la « parité *impaire* » (ou : *imparité*), on ajouterait un zéro supplémentaire à 1010100 pour obtenir 10101000, et un 1 à 0101101 pour aboutir à 01011011.

En liaison série asynchrone, c'est le code ASCII qui est aujourd'hui le plus communément utilisé ; presque systématiquement, on lui associe un bit de parité, ce qui fait qu'en pratique, on transmet des mots de 1 octet, comme on l'a vu antérieurement (il s'y ajoute les bits de *start* et de *stop*). Il faudra donc, à l'émission, ajouter un 8^{e} bit, et à la réception, compter les bits d'un mot pour vérifier leur parité ; si celle-ci n'est pas conforme, on déclenchera un processus d'erreur dont la première action pourra être une demande de ré-émission du mot incorrect. On examinera plus amplement les principes de la détection d'erreurs dans un chapitre spécial.

TABLEAU IV — LE CODE ASCII

Bits en positions 1, 2, 3 et 4		Bits en positions 5, 6 et 7							
		000	100	010	110	001	101	011	111
		0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	‘	p
1000	1	SOH	DC1	!	1	A	Q	a	q
0100	2	STX	DC2	”	2	B	R	b	r
1100	3	ETX	DC3	#	3	C	S	c	s
0010	4	EOT	DC4	$	4	D	T	d	t
1010	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
1110	7	BEL	ETB	’	7	G	W	g	w
0001	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
0101	10	LF	SUB	*	:	J	Z	j	z
1101	11	VT	ESC	+	;	K	[	k	{
0011	12	FF	FS	,	<	L	\	I	i
1011	13	CR	GS	–	=	M	]	m	}
0111	14	SO	RS	.	>	N	^	n	~
1111	15	SI	US	/	?	O	–	o	DEL

TABLEAU V — ABRÉVIATIONS INTERVENANT DANS LE TABLEAU DU CODE ASCII

NUL — Absence de caractère, blanc, espace.

SOH — « Start of Heading » : début d'en-tête.

STX — « Start of Text » : début du texte (et donc aussi fin de l'en-tête).

ETX — « End of Text » : fin du texte, commencé avec STX.

EOT — « End of Transmission » : fin de transmission.

ENQ — « Enquiry » : demande. Peut être utilisé pour demander au destinataire de s'identifier.

ACK — « Acknowledge » : accusé de réception.

BEL — « Bell » : sonnette.

BS — « Backspace » : marche arrière de 1 caractère.

HT — « Horizontale Tabulation » : tabulation horizontale.

LF — « Line Fed » : retour à une nouvelle ligne.

VT — « Vertical Tabulation » : tabulation verticale.

FF — « Form Fed » : passage à une page suivante.

CR — « Carriage Return » : retour du chariot.

SO — « Shift Out » : le code suivant est hors des caractères standards. Se termine par « Shift In ».

SI — « Shift In » : retour aux caractères standards du code utilisé.

DLE — « Data Link Escape » : change la signification d'un ou plusieurs caractères suivants. Sert à fournir des commandes supplémentaires.

DC1, DC2, DC3 et DC4 — « Device Controls... » : caractères de commande de terminaux ou périphériques.

NAK — « Negative Acknoledgment » : réponse négative à une question.

SYN — « Synchronous/Idle » : caractère de synchronisation ; lorsqu'aucun caractère n'est transmis, il peut être émis continûment.

ETB — « End of Transmission Block » : fin de transmission d'un bloc de données.

CAN — « Cancel » : annulation de la donnée précédente.

EM — « End of Medium » : fin physique de la carte, bande, ou autre support utilisé.

SUB — « Substitute » : remplacement.

ESC — « Escape » : caractère de contrôle fournissant un moyen d'étendre les possibilités de codage.

FS — « File Separator » ;

GS — « Groupe Separator » ;

RS — « Record Separator » ;

US — « United Separator » :
caractères de séparation à utilisation optionnelle, mais hiérarchiques, de FS (le plus général) à US.

SP — « Space » : espace, ou blanc.

DEL — « Delete » : suppression.

6. CIRCUITS D'ÉMISSION

CONVERSION PARALLELE-SERIE

Dès lors qu'on travaille avec un ordinateur, les données apparaissent généralement en parallèle. Pour les émettre en série, il faut procéder à une conversion parallèle-série, opérée à l'aide d'un simple registre à décalage. A la réception, une conversion inverse pourrait se faire à l'aide d'un même dispositif.

La figure 6 donne le principe de la conversion parallèle-série. On utilise un registre de 10 bits, ici, en supposant que le mot à transmettre soit de 8 bits, avec en plus un bit de start, donc un zéro en tête, et un bit de stop, donc un 1 en fin de mot. On n'a pas représenté les dispositifs qui imposent ce 0 et ce 1 à chaque chargement. Ce dernier est exécuté en parallèle, l'octet de données provenant de l'ordinateur étant logé dans le registre à décalage. Dès lors, une horloge va commander les décalages successifs, les bits sortant en série sur la ligne via un flip-flop servant de tampon.

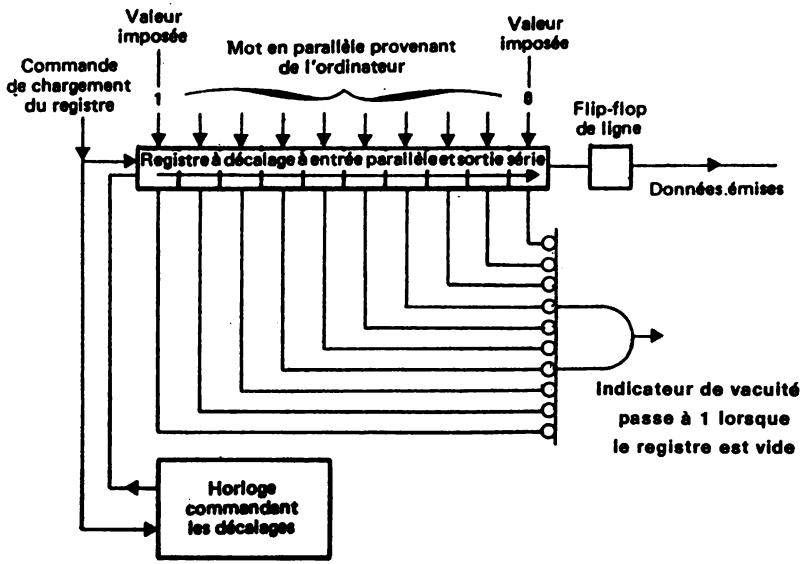

Valeur imposée
Mot en parallèle provenant de l'ordinateur
Valeur imposée
Commande de chargement du registre
1
8
Flip-flop de ligne
Registre à décalage à entrée parallèle et sortie série
Données émises
Indicateur de vacuité : passe à 1 lorsque le registre est vide
Horloge commandant les décalages

Fig. 6 — Principe de la conversion parallèle-série, à l'émission. Elle est effectuée par un simple registre à décalage.

A chaque décalage, un nouveau zéro est introduit à l'extrême gauche. Quand le dernier 1 (du *stop*) est sorti du registre, ce dernier ne contient plus que des zéros, état détecté par une porte à 10 entrées ; sa sortie passe alors à 1, niveau qui sert d'*indication de registre vide*, et en informe l'ordinateur qui pourra procéder à l'envoi du caractère suivant. S'il n'y a pas d'autre caractère à émettre, le *flip-flop de ligne* maintient la ligne au niveau 1.

7. PRINCIPE DES CIRCUITS DE RÉCEPTION

CONVERSION SERIE-PARALLELE

A la réception, une conversion inverse est exécutée selon le même principe, c'est-à-dire à l'aide d'un registre à décalage chargé en série et vidé en parallèle. Divers circuits de logique organisent les opérations, dont le principe est le suivant.

DETECTION DU START

FAUX DEPART

Un premier circuit détecte l'arrivée du bit de start, c'est-à-dire d'une transition du niveau *haut* au niveau *bas*. Un « timer » temporise alors pendant la durée de *la moitié d'un bit*, et l'on procède à une vérification : le niveau reçu est-il toujours *bas* ? Si oui, on peut présumer qu'un bit de start est bien en

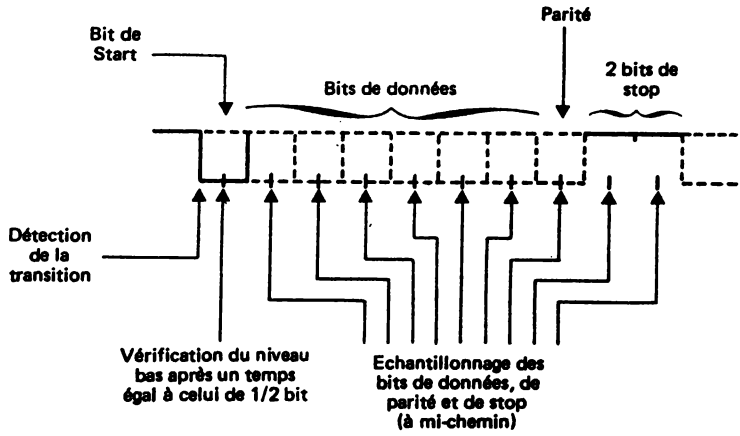

Parité
Bit de Start
2 bits de stop
Bits de données
Détection de la transition
Vérification du niveau bas après un temps égal à celui de 1/2 bit
Echantillonnage des bits de données, de parité et de stop (à mi-chemin)

Fig. 7 — A la réception du signal série, on commence par s'assurer que c'est bien un bit de start qui a été émis, puis on échantillonne les autres bits à mi-temps de leur durée fixée par convention.

cours de réception. Si non, c'est qu'on a eu affaire à une fausse alerte (un parasite, par exemple), ce qu'on appelle un « faux départ ». Cette vérification valide les circuits de réception qui, à partir de ce demi-bit (en temps), vont échantillonner les bits suivants à des intervalles de temps de 1 bit afin de « tomber » en plein milieu des créneaux (fig. 7). Le registre à décalage est chargé en série.

On traitera, par ailleurs, des circuits d'intendance qui interviennent également (création et vérification de la parité, comptage, détection du stop...).

8. LES BOUCLES 20 ET 60 mA

La liaison par « boucle de courant 20 mA » est restée très employée, en raison de son antériorité qui remonte à l'ère des liaisons « mécaniques ». On l'utilise parfois, par exemple, pour relier un ordinateur à une télé-imprimante ; la ligne fonctionne alors à faible vitesse de transmission, 110 bits par seconde.

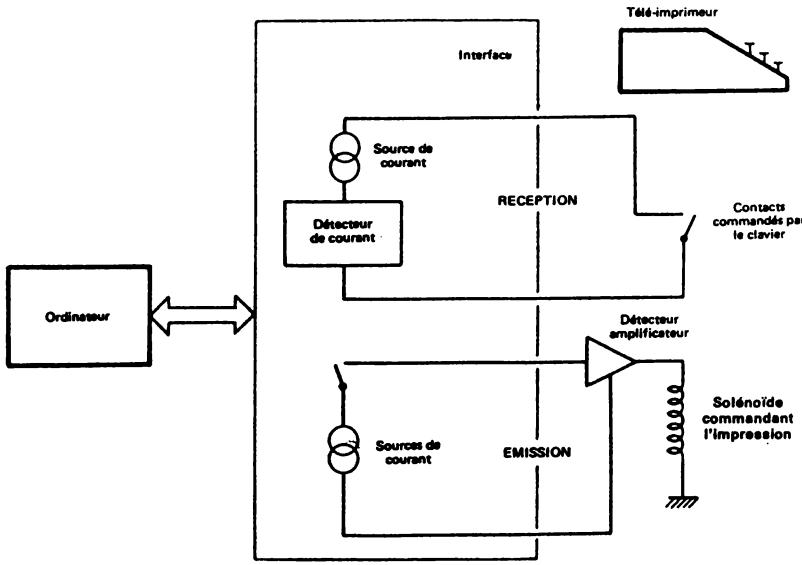


Fig. 8 — Principe d'une liaison ordinateur à télé-imprimeur, en série et asynchrone.

Le principe en est illustré figure 8. L'ordinateur asservit une interface, reliée au télé-imprimeur par une voie double, émission (*vers* l'imprimante) et réception (*de* l'imprimante). Lorsqu'on frappe un caractère sur l'imprimante, la combinaison des contacts qui se ferment et qui sont explorés en série est détectée par l'interface et transmise à l'ordinateur. Dans le sens inverse, ce dernier commande l'envoi d'un caractère qui excitera un relais de l'imprimante.

BOUCLE 20 mA

On s'arrange pour que la ligne soit parcourue par un courant de 20 mA. Pour cela, on crée une source de courant, à partir de la tension d'alimentation utilisée, en appliquant purement et simplement la loi d'Ohm. Si la tension disponible est de 20 V, on obtient 20 mA en insérant une résistance de 1000 Ω en série, à la condition que la ligne soit courte. Si la ligne est longue, on tient compte de sa résistance propre (de l'ordre de 10 à 15 Ω par cent mètres, par exemple).

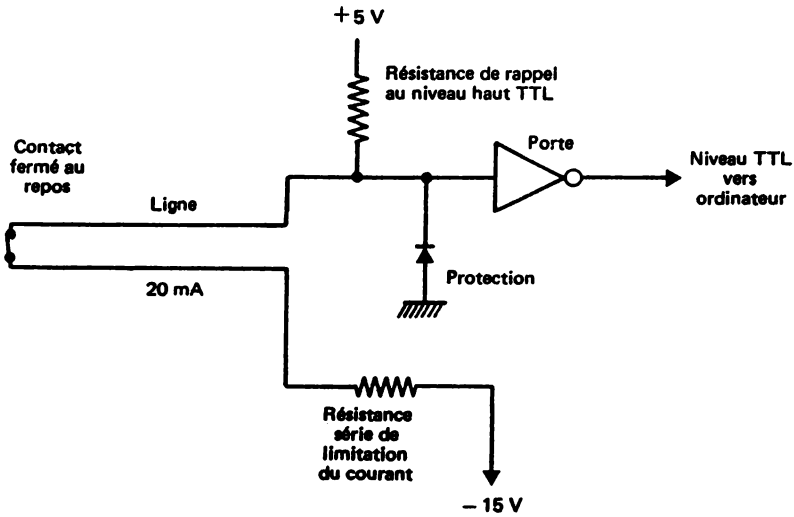


Fig. 9 — Exemple de circuit pour boucle de courant 20 mA.

De toutes façons interviennent également les résistances des contacts. L'aspect de la ligne peut ainsi revêtir l'allure de la figure 9, où l'on a supposé disposer des tensions -15 et $+5$ V, ce qui permet simplement de convertir les niveaux en niveaux compatibles TTL.

Il est évident qu'on obtient les mêmes 20 mA à partir d'une tension de 80 V et d'une résistance inférieure à 4 kΩ, ou avec toute autre combinaison. On touche du doigt l'un des défauts du principe, qui ne s'est pas prêté à une normalisation ; par contre, il est remarquablement simple et économique. Par ailleurs, la possibilité d'utiliser les tensions les plus diverses ne facilite pas le filtrage et l'usage de lignes longues sera génératrice de diaphonie.

Dans la première figure, on a situé les générateurs de courant dans l'interface. Celle-ci peut, à son tour, être logée dans l'ordinateur ou dans l'imprimante. Selon les cas, les générateurs apparaîtront alors :

— dans le terminal, le télé-imprimeur, qui est alors dit *actif*, l'ordinateur étant *passif* ;

— dans l'ordinateur, qui devient *actif*, l'imprimante étant alors dite *passive*.

Outre la boucle 20 mA, on a aussi utilisé des boucles 60 mA. Tout autre intensité est concevable.

BOUCLE 60 mA

Un montage pratique de boucle de courant commandant un télé-imprimeur est donné par la figure 10. Proposé par la société *Intel* (et installé sur la carte du micro-ordinateur d'évaluation SDK-85), il fait appel à un microprocesseur 8085 dont on utilise l'entrée série SID et la sortie série SOD. Dans chaque sens, lecture et écriture, est inséré un transistor tampon. Les échanges sont assurés à la vitesse de 110 bauds et les mots créés par le logiciel. Un caractère comporte 1 bit de start, 7 de donnée, 1 de parité et 2 de stop. Ici, les valeurs adoptées pour les résistances montrent qu'il s'agit d'une boucle 60 mA au clavier, mais bien moindre à l'impression puisque la tension différentielle est de 15 V pour 500 Ω de résistances environ.

COMMANDE DE TELEIMPRIMEUR PAR BOUCLE 20 mA

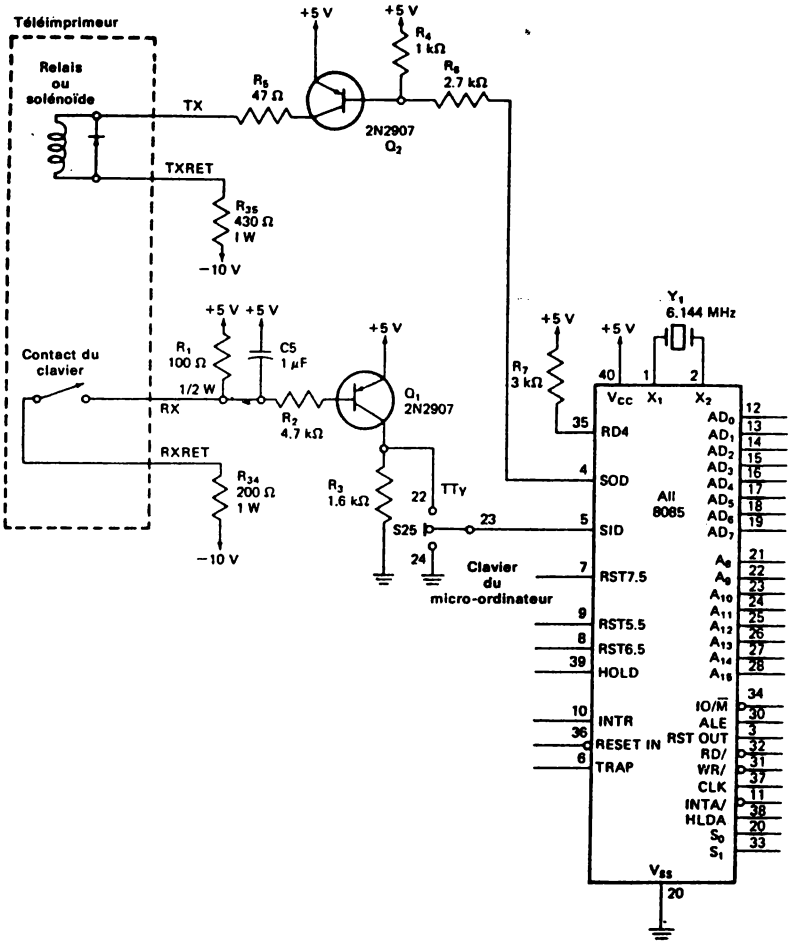


Fig. 10 — Commande d'un télé-imprimeur par un microprocesseur 8085 sur une boucle de courant.

9. LIAISON PAR COUPLEUR OPTIQUE

L'avènement des coupleurs optiques a permis de pallier l'un des inconvénients de la boucle 20 mA, due à l'absence de normalisation des tensions.

COUPLEUR OPTIQUE

Un coupleur optique est un composant constitué par une diode émettrice, ou LED (de « Light Emitting Diode ») et par un photorécepteur, le plus souvent un photo-transistor. La LED émet un faisceau lumineux qui porte l'information par sa présence ou son absence, et le phototransistor le détecte. Ainsi, il n'y a plus aucune liaison galvanique (par fils directs) entre l'émetteur et le récepteur, qui peuvent désormais travailler sous des tensions largement différentes sans risque.

Un exemple d'une liaison par coupleur optique est dessiné figure 11. L'ensemble LED et phototransistor est contenu dans un boîtier unique et se présente comme un simple composant. La Zener de gauche limite la tension qui attaque la LED et la protège contre une inversion accidentelle de cette tension ; les niveaux détectés sont amplifiés par un transistor et, selon les besoins, on peut obtenir une plus ou moins grande puissance de sortie.

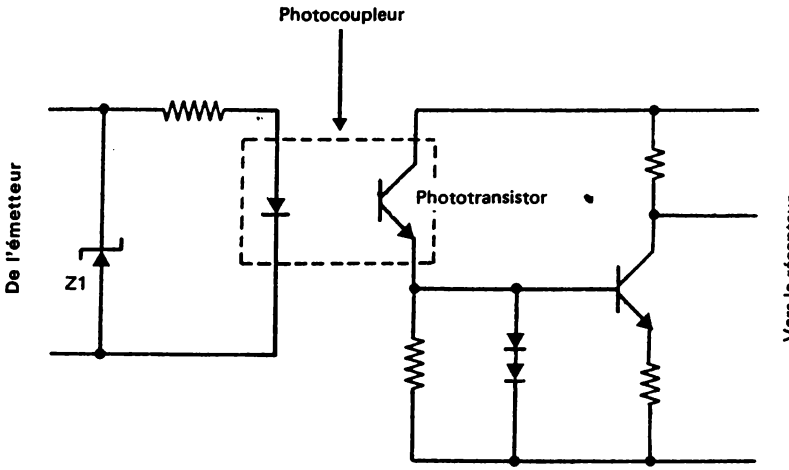


Fig. 11 — Principe d'une liaison par opto-coupleur. L'émetteur et le récepteur peuvent travailler sous des tensions largement différentes puisqu'il n'y a plus de liaison galvanique entre eux.

On remarquera que, dans le cas d'une liaison par opto-coupleur, il a fallu deux sources de courant : l'une du côté de l'émetteur, et l'autre au récepteur.

Ce type de liaison est généralement réservé à de courtes distances, par exemple moins du kilomètre. Au-delà, on utilisera un modem. Il est à noter qu'en variante de la liaison par fil mais toujours par couplage optique se développeront certainement de plus en plus des liaisons par *câble optique*.

10. LIAISONS EN V.24 et RS-232

La boucle de courant 20 mA est un mode de liaison qui « date ». D'autres méthodes ont été développées et normalisées, et par exemple et surtout celles appelées V.24 et RS-232. En raison de leur importance, elles font l'objet d'un chapitre distinct, le chapitre suivant.

11. UART ET USART

Un nom générique a été décerné aux interfaces « universelles » pour liaisons asynchrones, en circuit intégré, c'est celui d'« *Universel Asynchronous Receiver-Transmitter* », soit UART en abrégé. Elles sont alors proposées en un circuit intégré qui, très souvent, pourra assurer aussi bien des liaisons asynchrones que synchrones, d'où le nouveau sigle d'USART *(« Universal Synchronous-Asynchronous Receiver-Transmitter »)*. On présentera un tel circuit à l'issue du chapitre consacré aux liaisons synchrones en séries, mais on va examiner tout de suite la structure et le fonctionnement d'une UART type, appelée ACIA.

12. UNE INTERFACE ASYNCHRONE TYPE : L'ACIA

UART
ACIA

De nombreux fabricants réalisent des interfaces pour communications asynchrones en un seul circuit intégré. Tel est le cas de *Motorola* dont lecircuit ACIA, ou MC6850, est bien connu des utilisateurs du microprocesseur 6800. Le sigle ACIA provient de « *Asynchronous Communications Interface Adapter* ».

Ce circuit s'insère dans un système à microprocesseur comme le montre la figure 12. Il est relié via les bus-systèmes au 6800, et attaque un modem. Son synoptique simplifié est donné figure 13 et détaillé figure 14. Il est adressé comme une position mémoire par le microprocesseur et contient quatre registres, deux par sens de la liaison.

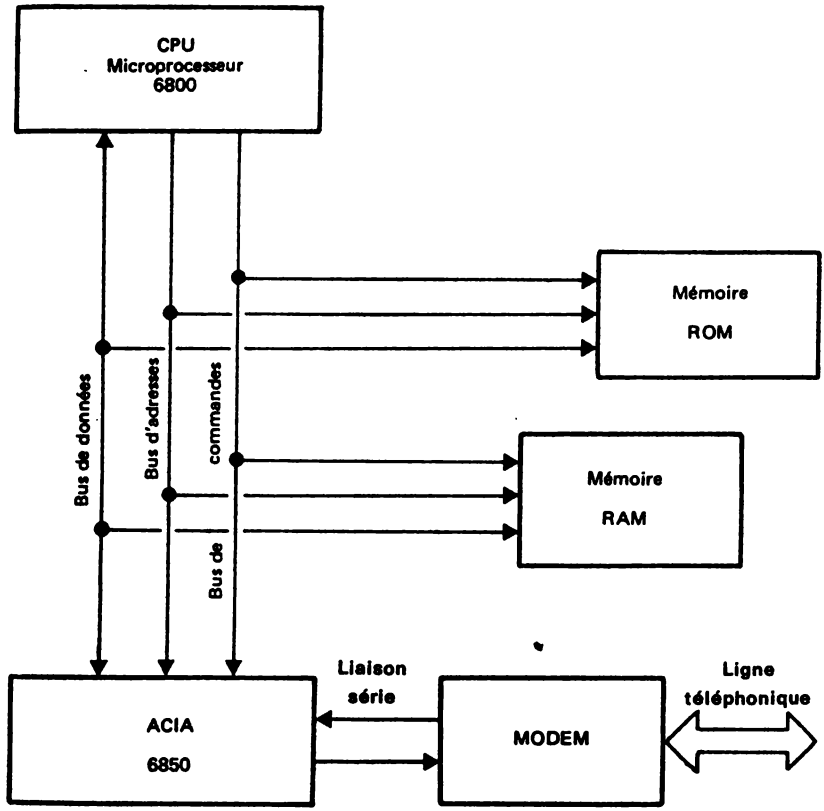


Fig. 12 — L'insertion de l'interface ACIA entre le modem et le microprocesseur, ici un 6800.

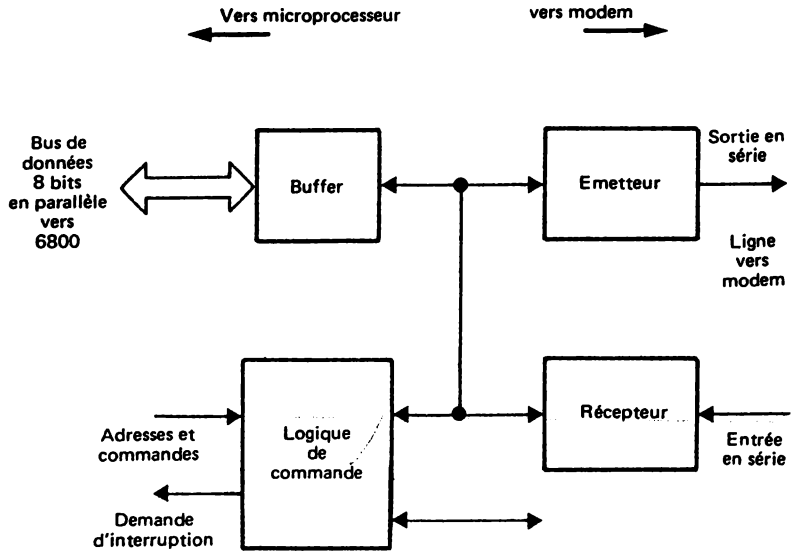


Fig. 13 — Synoptique très simplifié de l'interface.

Séquence d'émission. — Le registre de commande reçoit un mot de programmation qui définit les caractéristiques de la transmission : fréquence, longueur du mot, nombre de bits de stop, parité, etc. ; ce mot est chargé par le 6800 sous commande du programme.

Le caractère à émettre (du 6800 vers le modem) est chargé dans le *registre d'émission* puis transféré dans un *registre à décalage* qui exécute la conversion parallèle-série : l'entrée se fait en parallèle et la sortie en série résulte de huit décalages successifs. Cette sortie est précédée par l'émission d'un bit de start et suivie par un bit éventuel de parité ou d'imparité selon le mot de programmation, et par un ou deux bits de stop.

Avant chargement du registre d'émission, un *registre d'états* doit être lu : il précise si le registre d'émission est vide, donc disponible.

Réception. — Les données sont reçues dans un *registre de réception*. Le récepteur vérifie que le bit de start est bien arrivé en détectant sa première transition puis en testant que son niveau est resté constant après une durée équivalent à un demi-bit. En effet, si un parasite a pu donner un instant l'illusion qu'il s'agissait d'un bit de start, il aura disparu après ce temps de un demi-bit. Dès lors, l'horloge interne est synchronisée. Pour chaque caractère,

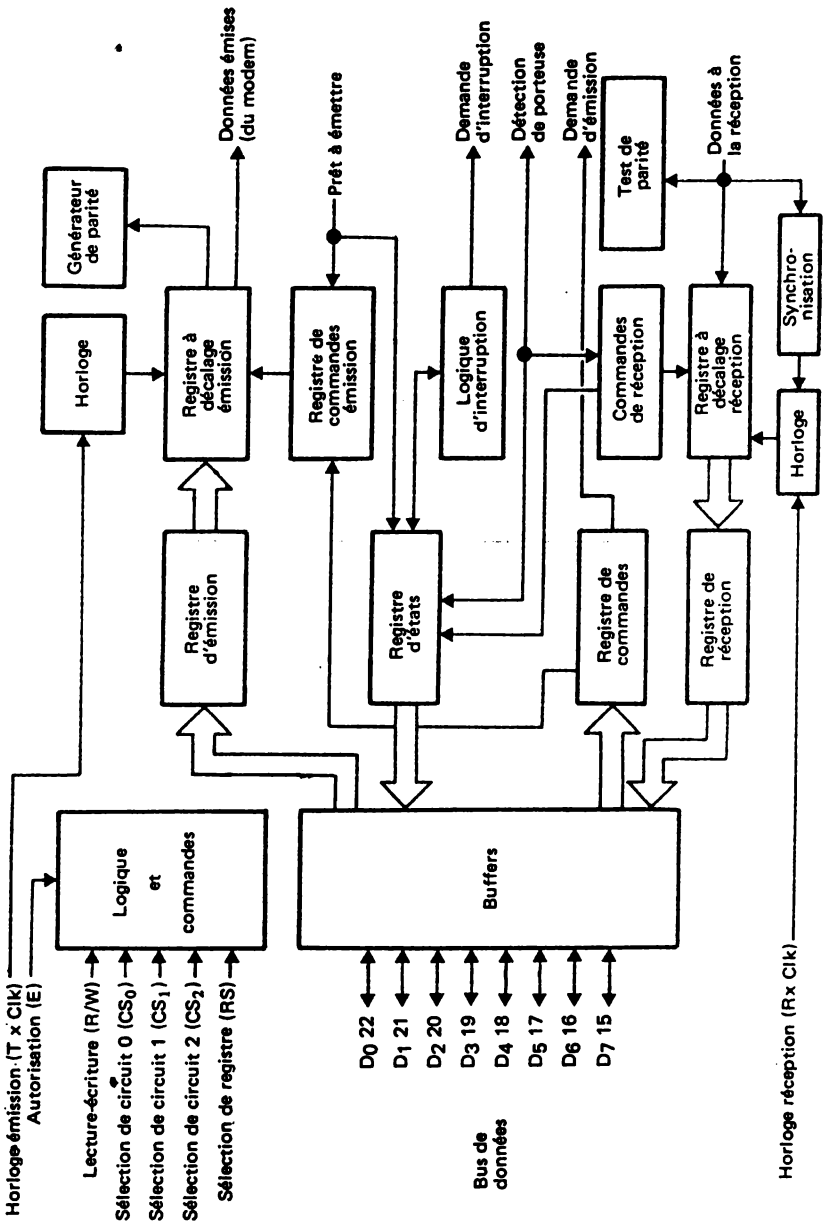


Fig. 14 — Synoptique détaillé de l'interface ACIA.

l'ACIA vérifie la parité et, s'il y a lieu, fournit une indication d'erreur au *registre d'états*. Ce dernier joue un rôle assez complexe puisqu'il fournit également des indications d'erreur de la trame d'un mot, ou de dépassement de vitesse (les mots surviennent trop rapidement) ; il témoigne aussi de l'état du registre de réception (vide ou plein).

Dans une séquence typique, le registre d'états est lu par le microprocesseur afin de déterminer si un caractère a été reçu d'un périphérique ; si oui, ce caractère est placé sur le bus de données dès que l'ordre de *lecture* est reçu du CPU (microprocesseur).

Signaux divers. — Le *bus de données* entre l'ACIA et le CPU est bidirectionnel, et du type trois états (le troisième état correspond à une déconnexion). Le *signal d'autorisation* (E) autorise le circuit à entrer en action ; il est dérivé de la phase 2 de l'horloge du système. La commande *lecture-écriture* (R/W) précise le sens de l'échange (*lecture* pour réception et *écriture* pour émission). Les signaux de *sélection de circuit* (CS), au nombre de trois, servent à l'adressage de l'ACIA ; la ligne de *sélection de registre* (RS) choisit entre les registres de données à l'émission et à la réception.

D'autre part, une ligne de *demande d'interruption* (ou IRQ, de « Interrupt Request ») est utilisée pour interrompre le CPU soit lorsque le registre d'émission est vide (dès lors que les interruptions ont été autorisées), soit lorsque le registre de réception est plein. Dans les deux cas, le microprocesseur peut alors poursuivre l'émission (en envoyant le mot suivant) ou la réception (en acquérant le mot reçu), puis revenir ensuite à son programme interrompu.

Commande du modem.— Quelques lignes spécialisées servent à la commande du modem. Il s'agit des lignes *Prêt à émettre* du modem ou « Clear-to-Send » (CTS) ; *Demande d'émission* ou « Request-to-Send » (RTS), et *Détection de porteuse* ou « Data Carrier Detect » (DCD).

Performances.— L'ACIA peut ainsi assurer des liaisons jusqu'à 500 kilobits par seconde. Elle détecte le *faux départ* (sur une impulsion parasite), des *erreurs de trame* (le premier bit de stop n'a pas été détecté : il y a donc eu une erreur de synchronisation, de transmission ou une rupture de séquence), un *dépassement* indiquant qu'un ou plusieurs caractères reçus ont été perdus, une *erreur de parité*. Présentée en un circuit intégré classique à double rangée de connexions, 24 broches, elle est alimentée sous 5 V et consomme environ 500 mW.

CHAPITRE IV

LIAISONS SÉRIE : LES NORMES RS-232 ET V.24

Dans le chapitre précédent, on a étudié les principes des liaisons séries asynchrones. L'un des modes de liaison les plus courants se fonde sur les normes RS-232C et V.24, qui fonctionnent d'ailleurs en asynchrone et en synchrone. Leurs applications concernent la liaison à une imprimante, un traceur (une table traçante), un modem, etc.

Les mots-clés de ce chapitre

DTE : « data terminal equipment », 62
DCE : « data circuit termination », 62
RS-232 : norme de liaison série, 64
RS-232C : plus récente version de la norme 232, 64

V.24 : norme de liaison série semblable à la RS-232C, 64
RS-423 : norme de liaison série, 72
RS-422 : norme de liaison série, 72
V.11 : norme de liaison série, 72

V.10 : norme de liaison série, 72
UART à téléimprimeur : liaison série, 74
Imprimante en mode série, 77
Connecteurs pour imprimantes série : en RS-232C, 79

Protocoles pour imprimantes : le dialogue, 82
BUSY : protocole de dialogue pour imprimante, 83
XON/XOFF : protocole de dialogue pour imprimante, 83
ACK : protocole de dialogue pour imprimante, 85

Liaison série sur IBM-PC, 85
Imprimante série avec IBM-PC, 89
Macintosh : liaisons série, 91
Macintosh et imprimante, 93

1. LES DTE ET DCE

La figure 1 illustre, de façon très générale, ce que l'on entend par interfaçage : il s'agit de relier deux éléments (ordinateurs, périphériques, modems...) appelés respectivement :

EQUIPEMENTS
DTE
DCE

• DTE, pour *Data Terminal Equipment* : c'est l'ordinateur, ou un terminal, une imprimante...

• DCE, pour *Data Circuit Termination* : souvent, il s'agira d'un modem (mais par extension, ce pourra aussi être un périphérique).

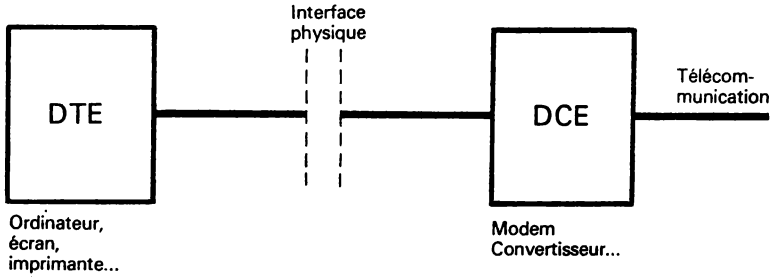


Fig. 1 — Organisation de principe (avec les termes consacrés, en télécommunication) d'une liaison.

Ce que l'on entend par « interface physique » sert alors à définir les quatre aspects de la liaison :

1) *Mécanique* (connecteurs, dimensions...) ;

2) *Electrique* (tensions, impédance, temps de montée du signal...) ;

3) *Fonctionnel* (fonctions, significations...) ;

4) *Procédures* (séquencement, horloge, tests...).

LES 25 LIGNES RS-232C

Broche du connecteur	Ligne	Direction : vers
1	Terre de protection	—
2	Données émises	DCE
3	Données reçues	DTE
4	Demande d'émission	DCE
5	Préparation d'émission	DTE
6	Données prêtes	DTE
7	Masse logique (commun)	—
8	Détection de porteuse	DTE
9	Réservée (pour test données)	—
10	Réservée (pour test données)	—
11	Non attribuée	—
12	Seconde détection de porteuse	DTE
13	Seconde préparation d'émission	DTE
14	Seconde transmission de données	DCE
15	Transmission horloge	DTE
16	Seconde « données reçues »	DTE
17	Réception horloge	DTE
18	Non attribuée	—
19	Seconde demande d'émission	DCE
20	Terminal prêt (DTR)	DCE
21	Détecteur qualité du signal	DTE
22	Détection sonnerie	DTE
23	Sélection de vitesse de transmission	DCE
24	Emission horloge	DCE
25	Non attribuée	—

Illustrons la notion de *procédure* par une liaison DTE (une imprimante, par exemple) - DCE (un modem) de la façon suivante. Le terminal (DTE) doit indiquer au modem (DCE) qu'il est *prêt* : sous tension, le papier est en place, etc. Pour cela, il active un circuit appelé DTR *(Data Terminal Ready)* : il s'agit d'un seul fil de liaison entre DTE et DCE qui, selon son niveau binaire, signifiera : « *Je suis prêt* » ou « *Je ne suis pas prêt* ». Il s'agit là du type de procédure qui appelle une normalisation. Où se situe ce fil ? Quel niveau témoigne de quoi ? Etc.

Afin de normaliser les liaisons en série, l'EIA *(Electrical Industry Association)* américaine a, en coopération avec les *Bell Laboratories*, émis les « recommandations » RS-232, dont la plus récente version est la 232C, alors que CCITT publiait l'« avis » V.24, très proche.

RS-232
RS-232C
V.24

Le RS-232C et le V.24 définissent un mode de *liaison série*, à l'origine destiné à relier un ordinateur et son modem. Par la suite, il s'est imposé entre ordinateurs et périphériques, par exemple ordinateur et écran, ou entre ordinateur et imprimante.

La liaison s'opère à l'aide de 25 fils maximum ; on pourrait penser que ce nombre suffirait pour une liaison parallèle mais, en fait, la plupart de ces lignes préparent ou servent la liaison série (c'est le cas de la ligne DTR, indiquant que le terminal est prêt, mais il y en a bien d'autres), certaines lignes restant même indéfinies. Leur liste est donnée dans la page précédente.

En fait, et au minimum, on peut n'utiliser que trois fils pour la liaison (fig. 2) : les fils 2 (émission), 3 (réception) et 7 (masse commune).

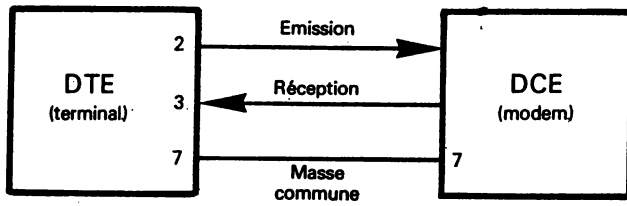


Fig. 2 — La liaison RS-232C peut s'effectuer sur trois fils, au minimum.

Une liaison complète, impliquant l'emploi d'une ligne téléphonique, pourrait être représentée comme le montre la figure 3.

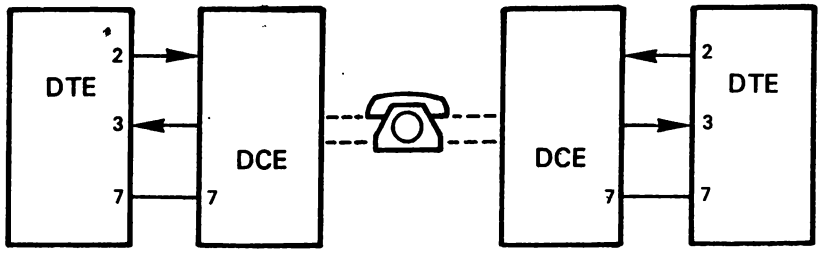


Fig. 3 — Exemple de liaison complète, via le réseau téléphonique.

En pratique, on utilisera souvent au moins les cinq signaux de la figure 4. En effet, aux trois lignes émission, réception, commun, vont venir s'ajouter deux lignes contrôlant l'échange dans ce mode qu'on définit « *en poignée de main* » :

- un fil *demande d'émission (« request to send »)*, le numéro 4 ;
- un fil *données prêtes (« data set ready »)*, le numéro 6.

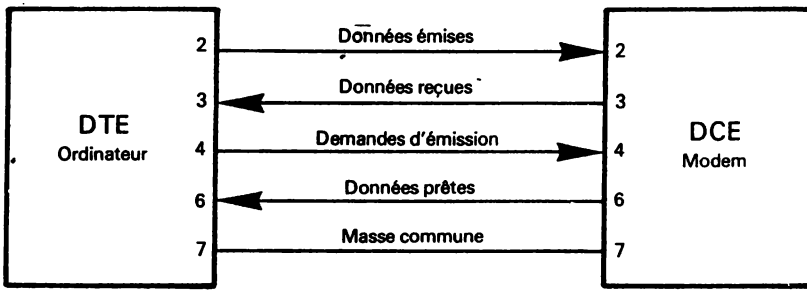


Fig. 4 — Les 5 fils de base d'une liaison.

Souvent, le DCE n'est pas un modem mais un périphérique et, par conséquent, on n'a plus affaire à un dispositif de transmission mais à un autre terminal. La liaison s'effectue *entre deux DTE* : il faudra croiser certaines connexions, les 2 et 3 pour le moins (fig. 5). En effet, *l'émission des données* (2) doit correspondre à *la réception* à l'autre bout. Ce sera généralement le câble de liaison, appelé *câble pour modem nul*, qui exécutera cette opération. D'autres fils, s'ils sont employés, devront subir le même sort (fig. 6).

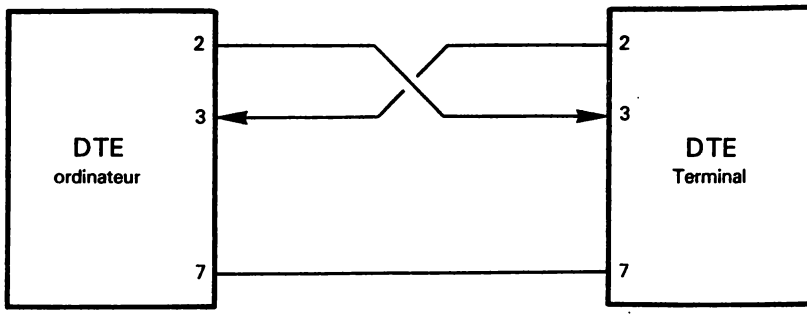


Fig. 5 — Si l'on a affaire à deux DTE, il faut croiser les fils 2 et 3.

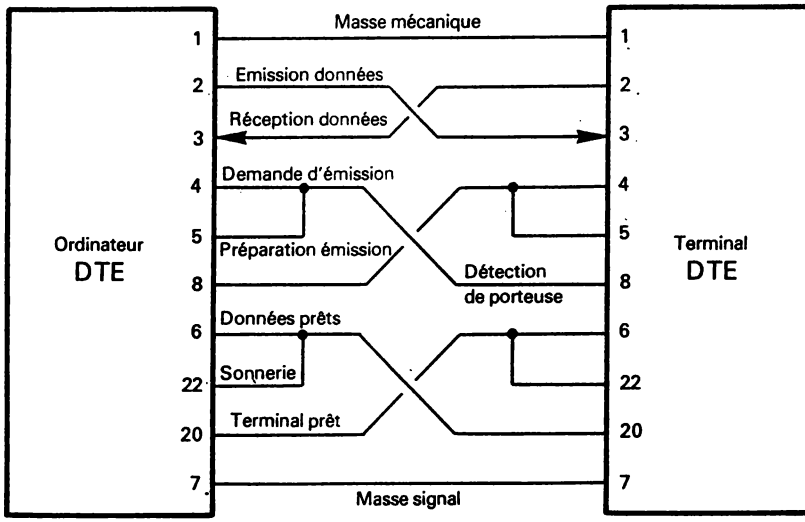


Fig. 6 — Le câble pour « modem nul » croise certaines connexions.

La définition des DCE et DTE peut sembler très arbitraire ; elle l'est, apparemment, car c'est le fabricant qui, grâce aux connecteurs et câbles fournis, leur attribuera le rôle choisi.

2. LES RS-232 ET V.24

Le connecteur normalisé, 25 broches, est dessiné dans la figure 7 ; on a donné l'affectation de certaines broches. On remarquera la présence de chiffres entre parenthèses : ils servent à désigner les circuits selon la norme V.24. La norme RS-232C, elle, emploie des lettres, ce qui constituera la principale différence ! Le tableau de la page 74 fournit les équivalencs.

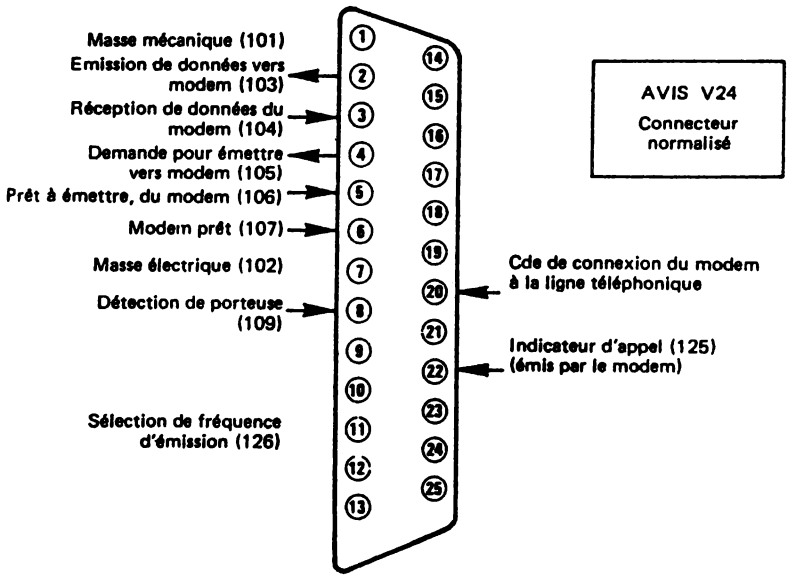

Masse mécanique (101)
Emission de données vers modem (103)
Réception de données du modem (104)
Demande pour émettre vers modem (105)
Prêt à émettre, du modem (106)
Modem prêt (107)
Masse électrique (102)
Détection de porteuse (109)
Sélection de fréquence d'émission (126)
Cde de connexion du modem à la ligne téléphonique
Indicateur d'appel (125) (émis par le modem)
AVIS V24
Connecteur normalisé

Fig. 7 — Le connecteur normalisé V.24 et quelques-unes des affectations de ses 25 broches.

L'organisation d'une dialogue type pourrait être la suivante :

1. Le terminal DTE met la ligne 108/2 (DTR) au niveau actif pour indiquer au modem (DCE) qu'il est prêt.

2. Le DCE met 107 (DSR) au niveau actif pour indiquer qu'il est prêt.

3. Le DCE rend 109 (RLSD) actif pour indiquer au terminal qu'il a détecté

une porteuse provenant de l'autre modem auquel il est connecté via le réseau téléphonique.

4. Le terminal rend 105 (RTS) actif pour demander au modem s'il peut envoyer les données.

5. Le modem met 106 (CTS) au niveau actif pour indiquer qu'il est prêt à émettre les données.

6. Le terminal commence à émettre les données vers le modem (sur 103) et à recevoir des données (sur 104).

En fait, ces protocoles et ces normes s'avèrent très complexes et sont loin d'être pleinement satisfaisantes, ce qui signifie qu'elles sont appelées à évoluer encore. Por plus de rigueur, on devrait d'ailleurs dire que la norme RS-232C additionne des fonctions des normes CCITT V.24 et V.28 (électriques) ainsi que de ISO-2110 (connecteur).

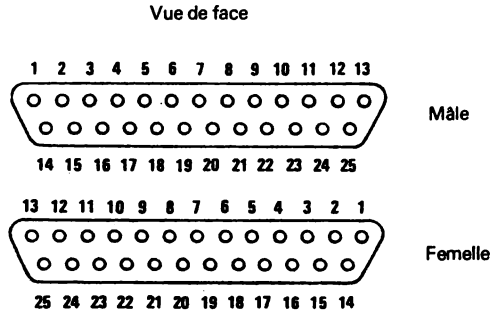


Fig. 8. — Le connecteur standard RS-232C.

La norme RS-232C est ancienne et utilise des niveaux logiques qui sont :

- à l'émission, de +5 à +15 V et de −5 à −15 V ; .
- à la réception, de +3 à +25 V et de −3 à −25 V.

C'est ce que montre la figure 9, où l'on constate que les niveaux « morts » vont de +5 à −5 V à l'émission, de +3 à −3 V à la réception. En logique ***négative***, les niveaux sont définis ainsi que le montre la figure.

L'excursion maximale différentielle est donc de 50 V, ce qui explique qu'une liaison RS-232C ait une portée de près de 20 mètres. Ce sera le cas, par exemple, entre un ordinateur et une imprimante ; on comparera cette distance aux 3 m admis par une liaison parallèle où les niveaux logiques sont ceux de la famille TTL (0 et 5 V).

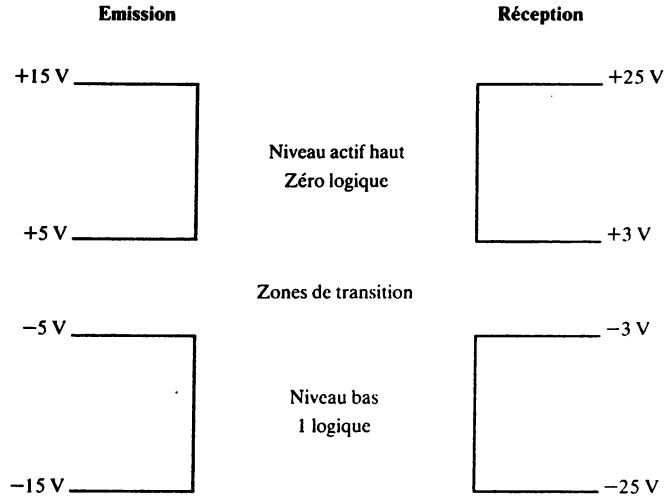


Fig. 9 — Les niveaux normalisés RS-232C, à l'émission et à la réception.

Les choses se compliquent cependant dans la mesure où ces niveaux (± 25 V) utilisent la *logique négative*, mais pour les données uniquement, car les lignes de contrôle recourent à la logique positive (avec une tension positive pour le 1 et négative pour le 0).

La liaison peut s'effectuer en *mode série asynchrone*, avec bits de start, de stop et de parité (éventuellement), comme on l'a vu dans le chapitre définissant ce mode. La vitesse de transmission varie de 50 bits à 9600 bits par seconde en passant par des valeurs intermédiaires typiques de 75, 110 ou 134,5, ou 150, 300, 600, 1200, 1800, 2400, 3600 et 4800 bits par seconde. Divers codes de transmission peuvent être employés, mais le plus usuel est le code ASCII, sur 7 bits ou 8 bits.

La figure 8 montre l'organisation du connecteur RS-232C et la numérotation de ses broches. On pourra utiliser les connecteurs de références DB-25 d'*ITT-Cannon, TRW-Cinch*, ou 17-81250 et 17-91250 d'*Amphenol*, etc.

BROCHAGE DU CONNECTEUR EN V.24 et RS-232C

Broche	Norme		Source		Désignation		
	V.24 CCITT	RS-232C EIA	DTE	DCE	Américaine	Abrégé	Française
1	101	AA			Earth ground	—	Terre
2	103	BA	•		Transmitted Data	TD	Emission données
3	104	BB		•	Received Data	RD	Réception données
4	105	CA	•		Request to Send	RTS	Demande d'émission
5	106	CB		•	Clear to Send	CTS	Préparation émission
6	107	CC		•	Data Set Ready	DSR	Données prêtes
7	102	AA			Logic ground	—	Masse signaux
8	109	CF		•	Carrier Detect	RLSD	Détection porteuse
9	—	—			Reserved	—	Réservée
10	—	—			Reserved	—	Réservée
11	126	—	•		Unassigned	—	Non affectée
12	122	SCF		•	Secondary Carrier Detect	RLSD	Seconde détection porteuse
13	121	SCB		•	Secondary Clear to Send	CTS	Seconde préparation émission
14	118	SBA	•		Secondary Transmitted Data	TD	Seconde émission données
15	114	DB		•	Transmit Clock	—	Emission horloge

BROCHAGE DU CONNECTEUR EN V.24 et RS-232C (suite)

Broche	Norme		Source		Désignation		
	V.24 CCITT	RS-232C EIA	DTE	DCE	Américaine	Abrégé	Française
16	119	SBB		•	Secondary Received Data	RD	Seconde réception données
17	115	DD		•	Receiver Clock	—	Réception horloge
18	141	—	•		Unassigned	—	Non affectée
19	120	SCA	•		Secondary Request to Send	RTS	Seconde demande émission
20	108/1	—	•		Data Terminal Ready	CDSL	Terminal prêt
	108/2	CD	•			DTR	
21	140	—	•		Signal Quality Detect	—	Détection qualité signal
22	125	CE		•	Ring Detect	RI	Détection sonnerie
23	111	CH	•		Data Rate Select	—	Sélection vitesse transmission
24	113	DA	•		Transmit Clock	—	Emission horloge
25	142	—		•	Unassigned	—	Non affectée

NORMES EIA COMPARÉES

Caractéristiques	RS 232 - C	RS 423	RS 422
Mode pour les données	Un fil	Un fil	Différentiel
Longueur max du câble pieds mètres	 50 17	 2 000 666	 4 000 1 333
Débit max (kilobauds)	20	300	10 000
Tension max de sortie en circuit ouvert (V)	± 25	± 6	6 en différentiel
Tension min de sortie en charge (V)	± 5 à ±15	± 3,6	2 en différentiel
R de sortie min (Ω) **ou courant** (μA)	Ro = 300 Ω	100μA de − 6 à + 6 V	100μA de + 6 à 0,25 V
Courant max de court-circuit (mA)	± 500	± 150	± 150
Seuils max du récepteur (V)	− 3 à + 3	− 0,2 à + 0,2	− 0,2 à + 0,2
Tension max à l'entrée récepteur (V)	− 25 à + 25	− 12 à + 12	− 12 à + 12

3. NORMES RS-423 ET 422 (V.11 ET V.10)

RS-423
RS-422
V.11
V.10

La norme RS-232 est relativement ancienne et, comme le montre le tableau ci-dessus, elle s'applique à une liaison sur un fil pour les données, à moins de 20 kilobauds et sur une distance maximale de 50 pieds. Aussi, en 1976, l'*EIA* a-t-elle édicté deux nouvelles normes qui sont la RS-423, sur un fil, et la 422, différentielle ; cette dernière exige donc des drivers différentiels.

Pour définir l'ensemble des fonctions et les caractéristiques mécaniques de l'interface, l'*EIA* a publié, en 1977, la norme RS-449.; elle décrit un total de 30 circuits d'interfaçage et des connecteurs 37 broches et 9 broches.

Le *CCITT* a, de son côté, émis les recommandations V.10 et V.11 respectivement équivalentes aux RS-422 et RS-423, et les V.24 et V.54 compatibles avec la RS-449.

4. QUELQUES MONTAGES TYPES

La figure 10 donne l'exemple d'une liaison directe la plus dépouillée, terminal à ordinateur, donc sans modem, en RS-232C.

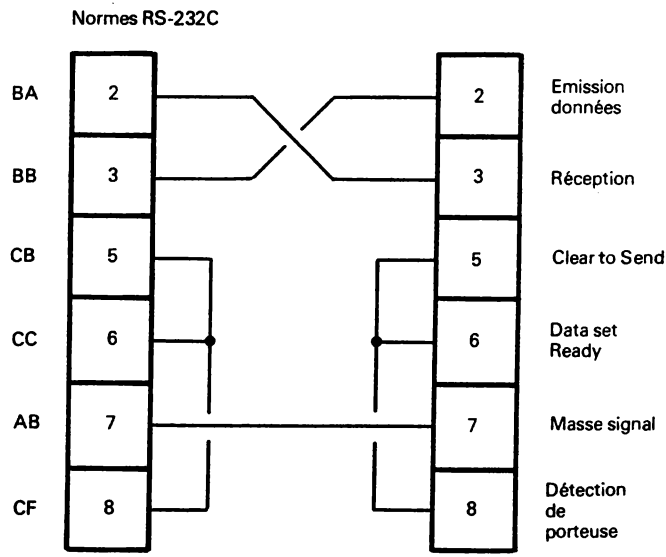


Fig. 10 — Liaison directe type ordinateur à terminal.

UART A TELEIMPRIMEUR

Supposons que la sortie de l'ordinateur se fasse sur un UART *(émetteur-récepteur universel asynchrone)* ; la connexion la plus simple se fera à l'aide des montages de la figure 11, respectant les niveaux normalisés. Un tel montage peut être développé comme le montre la figure 12, cette fois avec une liaison UART à téléimprimeur ; le sigle TTY provient de « Teletype », marque célèbre de téléimprimeur ; Tx signifie transmetteur et Rx, récepteur.

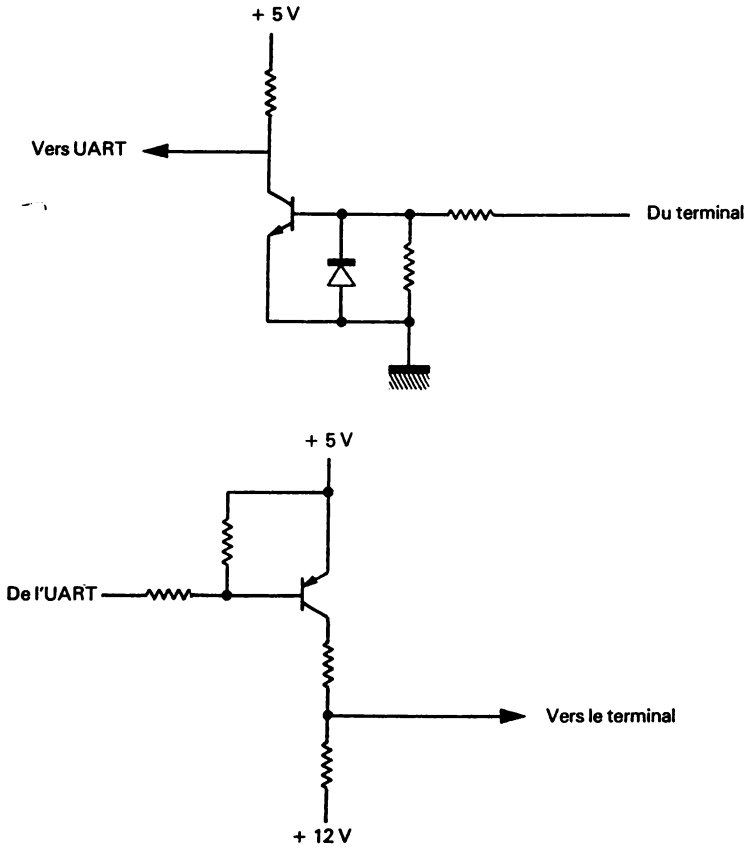


Fig. 11 — Liaison des données UART à terminal.

On pourrait aussi passer par des isolateurs opto-électriques (fig. 13), ou par des circuits rapides en logique bipolaire Schottky pour accroître la vitesse.

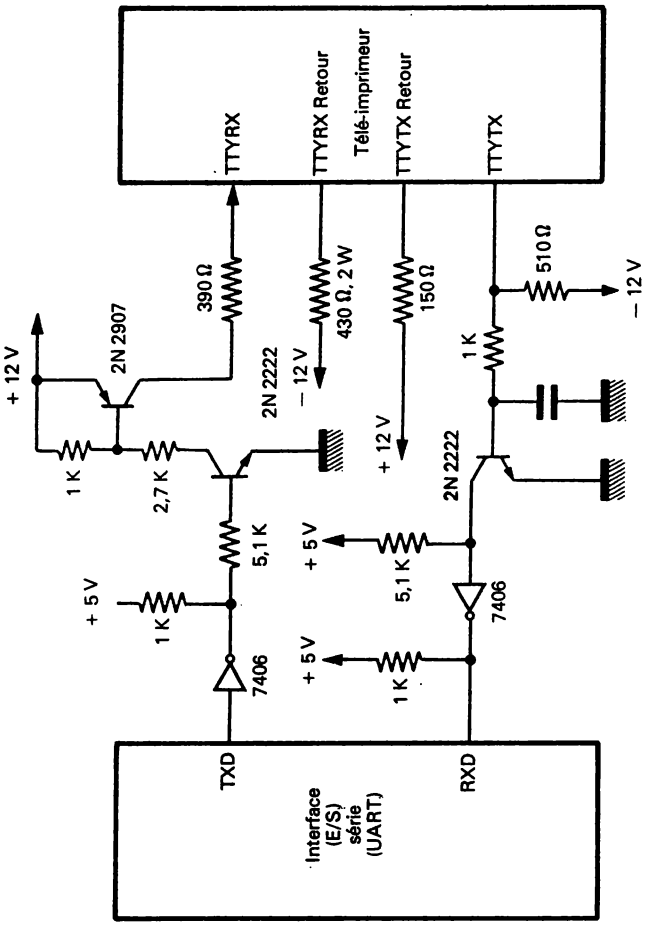


Fig. 12 — Liaison RS-232C à un téléimprimeur.

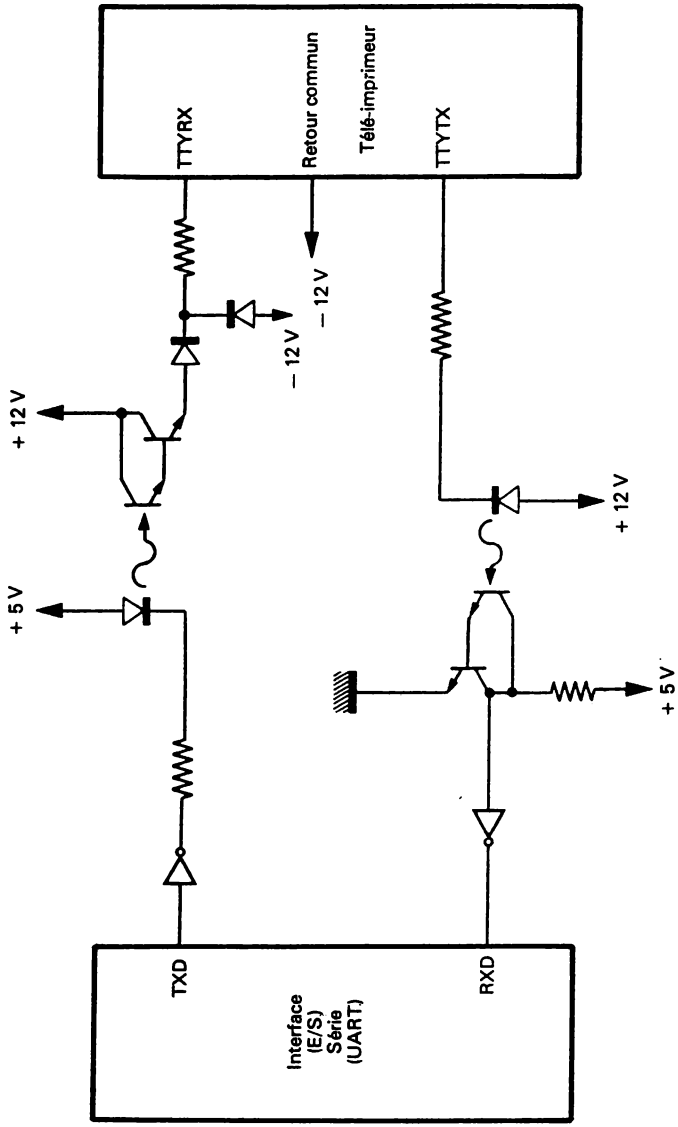


Fig. 13 — Liaison à un téléimprimeur avec coupleurs optiques et séparation des masses et alimentations.

5. PRINCIPE DES ÉCHANGES EN MODE SÉRIE ASYNCHRONE AVEC UNE IMPRIMANTE

IMPRIMANTE EN MODE SERIE

L'exemple type d'une liaison série asynchrone est celle entre l'ordinateur et une imprimante. Dans ce cas, et si la norme de base est bien respectée, elle est aussi souvent « aménagée » ; ainsi, nombre de fabricants adjoignent des fils pour liaison par *boucle de courant* et, de ce fait, l'organisation des connecteurs différera. Ils utiliseront une dizaine ou une quinzaine de fils.

De toutes façons, les impératifs sont les suivants. Il faut, bien sûr, transmettre les données en mode asynchrone mais, en outre, il est nécessaire de gérer l'échange, c'est-à-dire d'organiser le dialogue entre le micro-ordinateur et l'imprimante. Pour cela, il convient de définir (et ce, pour tous deux) :

• *La vitesse de transfert*, en *bits par seconde* ou en *bauds*, ce qui est, ici, équivalent. Elle va généralement de 150 à 19 200 bits par seconde.

La vitesse de transfert n'a rien à voir avec la vitesse de frappe de l'imprimante car celle-ci est généralement équipée d'un buffer d'entrée, stockant un certain nombre de caractères.

Si l'on suppose qu'un caractère occupe 10 bits (7 bits de données, 1 bit de start, 1 bit de parité, 1 bit de stop), les temps typiques sont les suivants :

TEMPS TYPIQUES

Vitesse (bits/seconde)	Durée d'un bit (ms)	Durée d'un caractère sur 10 bits (ms)
150	6,66	66,6
300	3,33	33,3
600	1,67	16,7
1 200	0,8	8,3
2 400	0,42	4,2
4 800	0,21	2,1
9 600	0,10	1
19 200	0,05	0,5

En fait, et avec un codage sur 8 bits de données et 2 bits de stop, on aboutit à un total de 12 bits par caractères, ce qui réduit sa vitesse de transmission.

• *La composition d'un mot asynchrone* : nombre de bits de donnée et d'arrêt, l'existence et le type de parité. Le bit de *start* est obligatoire.

• *Les types et niveaux des signaux* : niveaux logiques pour le 1 et le 0, tensions utilisées (généralement de ± 5 à ± 15 V).

• *Enfin, les procédures de dialogue* : elles sont nombreuses ; nous allons examiner les quatre plus courantes un peu plus loin.

6. CONNECTEURS POUR IMPRIMANTES SÉRIE

Le connecteur d'interface série répond généralement aux normes RS-232C ou V.24 mais est souvent « aménagé » ou modifié pour des exigences particulières. En fait, il n'existe pas de norme rigoureuse ici. Si les fils de base sont respectés, on y ajoute, selon les circonstances, des connexions pour boucle de courant ou autres.

CONNECTEURS POUR IMPRIMANTES SERIE

En mode RS-232C « pur », l'imprimante Pinwriter de *NEC* fait appel aux 9 lignes de la figure 14. Nous n'avons pas indiqué les numéros de broches, côté ordinateur, mais n'oubliez pas qu'avec une sortie série généralement prévue pour un modem, il faudra croiser certains fils, ce qu'on va illustrer plus loin avec une application à l'IBM-PC.

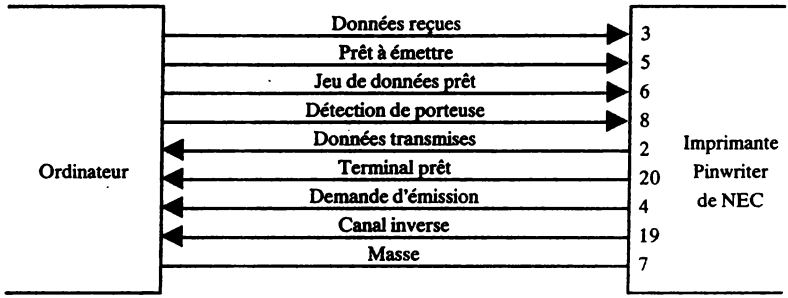


Fig. 14 — Liaison série asynchrone entre un ordinateur et une Pinwriter.

Le tableau suivant fournit quelques équivalences pour les références et les noms adoptés :

CABLE DE LIAISON SÉRIE A UNE PINWRITER

Broche du connecteur	Appellation		Nom du signal EIA
	EIA	CCITT	
2	BA	103	Transmitted Data
3	BB	104	Received Data
4	CA	105	Request To Send
5	CB	106	Clear To Send
6	CC	107	Data Set Ready
7	AB	102	Signal Ground
8	CF	109	Carrier Detect
19	SCA	120	Reverse Channel
20	CD	108	Data Terminal Ready

Les fonctions affectées à ces fils sont les suivantes :

FONCTION DES LIGNES POUR LA PINWRITER

Signal	De ou vers l'imprimante	Fonction
Détection de porteuse	Vers	Lorsqu'il est *haut*, la liaison est établie et utilisable pour l'imprimante.
Terminal prêt	De	Lorsqu'il est *haut*, l'ordinateur sait que l'imprimante est prête.
Donnée prête	Vers	Ordonne à l'imprimante de prélever les données.
Demande d'émission	De	Cette ligne est *haute* lorsque « Terminal prêt » et « Donnée prête » sont *hautes.*
Prêt à émettre	Vers	*Haute* lorsque l'ordinateur est prêt à envoyer des données.
Données transmises	De	Les bits de données, en mode série, avec leur accompagnement (start, stop, parité).
Canal inverse	De	Passe au niveau haut sur incident : • mémoire tampon pleine, • fin de papier, • erreur, • désélection, • etc.

Dans le cas où le fabricant a prévu des lignes supplémentaires pour *boucle de courant*, par exemple, on pourra aboutir à des connecteurs plus chargés, tel celui des SR 10/15 de *Star/Hengstler* dont les fils, vus du *côté imprimante* sur un connecteur DB-25 (25 broches), sont les suivants :

CONNECTEUR DES SR 10/15 STAR/HENGSTLER

Broche	Signal	Entrée/sortie (pour l'imprimante)	Fonction
1	GND	—	Masse mécanique
2	TXD	Sortie	Données transmises
3	RXD	Entrée	Données reçues par l'imprimante
4	RTS	Sortie	Demande d'émission
5	CTS	Entrée	Prêt à émettre
6	DSR	Entrée	Données prêtes
7	SG	—	Masse électrique
8	DCD	Entrée	Détection de porteuse
9	TTY TXDR		Retour de boucle de courant 20 mA pour données partant de l'imprimante
10	TTY TXD	Sortie	Départ de boucle de courant 20 mA pour données partant de l'imprimante
11	RCH	Sortie	Pour commande série Busy ou ON/OFF
12	NC	—	Non connecté, inutilisé
13	GND	—	Masse mécanique
14 à 16	NC	—	Inutilisés
17	TTY TXDR		Retour de boucle de courant 20 mA pour données partant de l'imprimante
18	TTY RXDR		Retour de boucle de courant 20 mA pour données entrant dans l'imprimante

(suite page suivante)

CONNECTEUR DES SR 10/15 STAR/HENGSTLER

(suite du tableau de la page précédente)

Broche	Signal	Entrée/sortie (pour l'imprimante)	Fonction
19	TTY RXD	Sortie	Sortie de boucle de courant 20 mA pour données entrant dans l'imprimante
20	DTR	Sortie	Terminal prêt
21 et 22	NC	—	Inutilisés
23	TTY RXDR		Retour de boucle de courant 20 mA pour données allant vers l'imprimante
24	TTY TXD	Sortie	Sortie 20 mA des données partant de l'imprimante
25	TTY RXD	Entrée	Sortie 20 mA pour données allant vers l'imprimante

Il y aura donc toujours lieu d'être prudent quant à l'usage des câbles pour liaisons série. Il nous reste à examiner comment est organisé le dialogue ordinateur-imprimante, pour les protocoles les plus usuels et, pour cela, on va se référer aux lignes ci-dessus, pour l'imprimante *Star*.

7. PROTOCOLES DE DIALOGUE AVEC L'IMPRIMANTE

PROTOCOLES POUR IMPRIMANTES

Les quatre principaux protocoles de dialogue ordinateur-imprimante en mode série sont le *Busy par caractère* (« *Busy* » signifiant « occupé »), le *Busy par bloc*, le *mode Xon/Xoff* et, enfin, le mode *Acknowledge* (acquisition). Les deux premiers font appel à deux fils spécialisés, broches 20 (DTR) et 11 (RCH) alors que les deux suivants recourent au fil de retour des données, de l'imprimante vers l'ordinateur (broche 2, TXD).

7.1. Mode BUSY par caractère

BUSY

En anglais, son nom complet est *Serial busy protocole (1 byte) mode*. Il recourt plus particulièrement aux lignes 20 (DTR) et 11 (RCH) qui passent au niveau *haut* lorsque l'imprimante est prête à recevoir des données, et reviennent au niveau *bas* pendant la transmission d'une donnée. Si la mémoire tampon (buffer) de l'imprimante est pleine, elles restent au niveau *bas*. Le chronogramme type en est donné figure 15*a*. En fait, ces lignes semblent se substituer à leur équivalent *Busy*, en mode parallèle.

7.2. Mode Busy par bloc

Le mode *Busy par bloc (« Serial busy protocole (1 block) mode »)* diffère du précédent par le fait que là, les lignes DTR et RCH ne passent à l'état OFF que lorsque le buffer de l'imprimante est presque plein (fig. 15*b*).

7.3. Mode Xon/Xoff

XON/XOFF

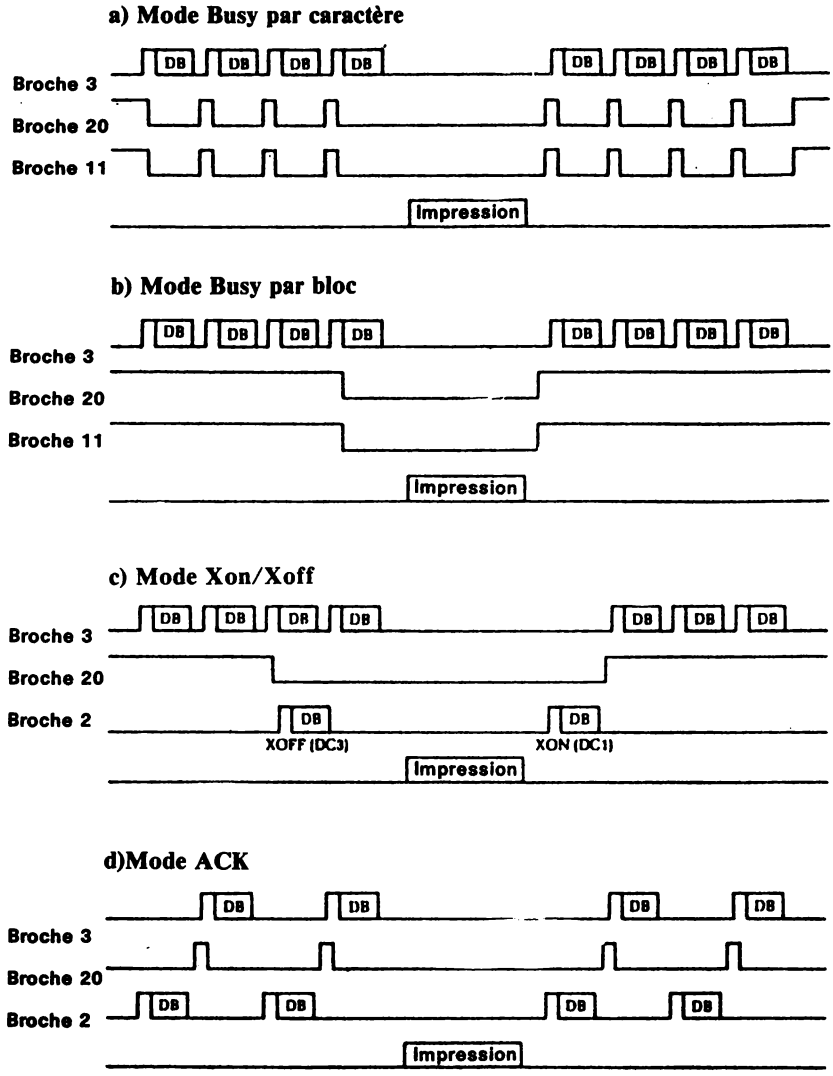
On a constaté que la ligne RXD servait à envoyer des données du micro-ordinateur à l'imprimante. Or, celle-ci dispose d'une ligne inverse, TXD (broche 2) permettant à l'imprimante d'émettre des caractères vers l'ordinateur. C'est elle qui va intervenir ici. Afin de bien comprendre ce mode, il faut encore savoir que :

Dans le code ASCII, il existe des caractères :

- **DC1, code 17 en décimal, ou Ctrl + Q ;**
- **DC3, code 19 en décimal, ou Ctrl + S.**

Ce sont ces codes qui vont intervenir sur la ligne TXD.

- **Le code DC1 est encore appelé *Xon*.**
  DC3 est alors appelé *Xoff*.



DB = « data byte », octet de donnée ou caractère

Fig. 15 — Les quatre principaux protocoles d'échange micro-ordinateur à imprimante en mode série asynchrone.

Lorsque la mémoire tampon de l'imprimante est presque remplie, celle-ci émet un *Xoff* sur la ligne TXD, vers l'ordinateur, afin d'interrompre la transmission des données. Lorsqu'elle peut en recevoir d'autres, elle émet un *Xon* sur la même ligne (fig. 15*c*).

7.4. Mode ACK

ACK

C'est le mode *Acknowledge*, accusé de réception, transmis également sur la ligne TXD.

La notation ACK est fournie par le code ASCII numéro 6, en décimal (Ctrl + F).

Il ne s'agit donc plus d'un fil spécifique, comme avec la liaison parallèle. Cet ACK est émis par l'imprimante vers l'ordinateur lorsqu'elle est prête à accepter des données (fig. 15*d*), caractère par caractère.

8. LIAISON SÉRIE SUR L'IBM-PC

LIAISON SERIE SUR IBM-PC

Sur l'IBM-PC, la liaison série asynchrone est assurée par une carte d'interface, pouvant fonctionner en mode RS-232C ou en boucle de courant ; son synoptique est donné figure 16. Son circuit essentiel est le 8250 de liaison asynchrone, capable de :

- ajouter ou supprimer (en émission ou réception) les bits de *start*, de *stop* et de *parité* ;
- travailler sur des mots de 5 bits, 6 bits, 7 ou 8 bits,
- en parité parie, impaire, ou sans parité ;
- avec 1 bit de stop, ou 1 1/2, ou 2 ;
- fonctionner du continu à 9600 bauds ;
- détecter le faux départ (par test après 1/2 bit).

En fait, on peut programmer toutes les vitesses de transfert, de 50 bits par seconde (ou *bauds*, ici) à 9600. Ce circuit fournit aussi des commandes pour un *modem* :

- CTS : *clear to send*, prêt à émettre ;
- RTS : *request to send*, demande d'émission ;
- DSR : *data set ready*, données prêtes ;
- DTR : *data terminal ready*, terminal de données prêt ;
- RI : *ring indicator*, sonnerie ;
- CD : *carrier detect*, détection de porteuse.

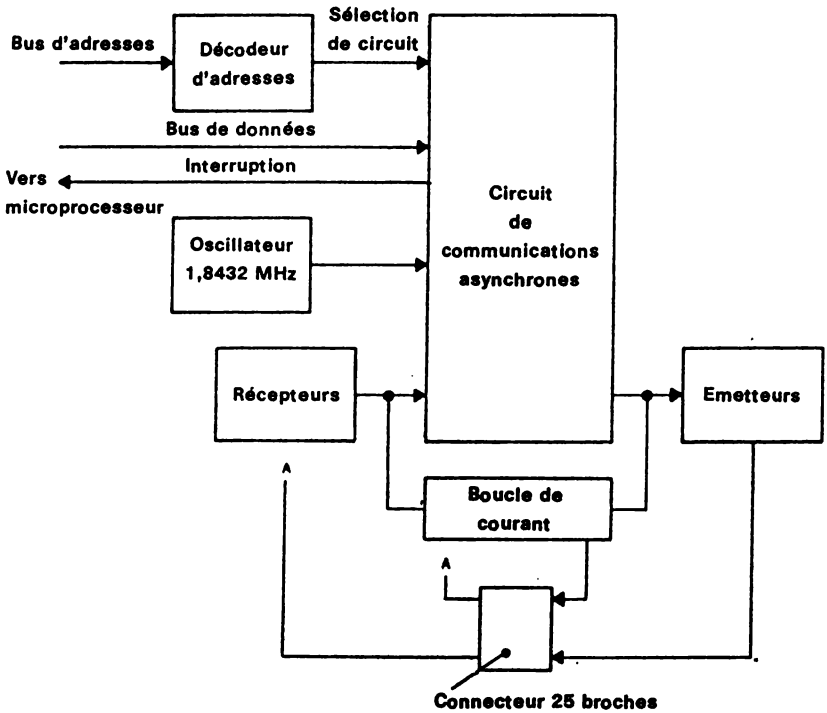


Fig. 16 — Synoptique de la carte d'interface de liaison série asynchrone de l'IBM-PC.

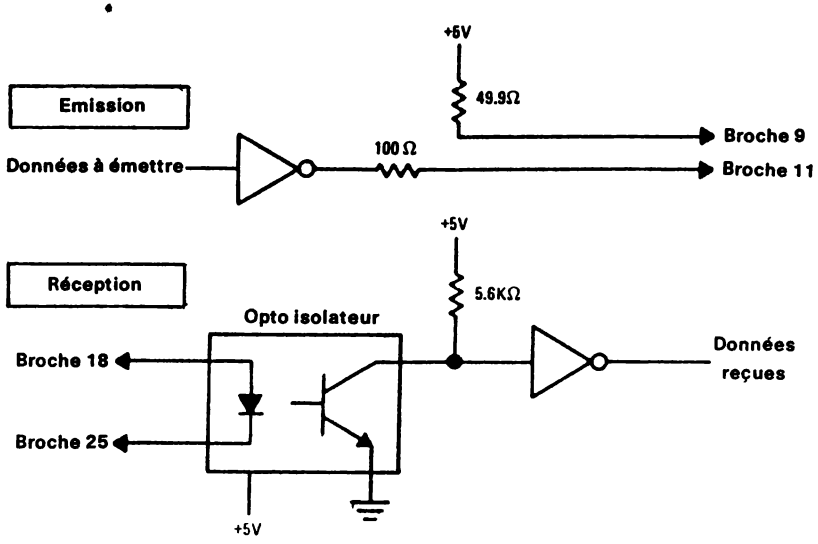


Fig. 17 — Principe de la boucle de courant 20 mA sur l'IBM-PC.

La carte d'interface débouche sur un connecteur mâle 25 broches, différent du connecteur normalisé RS-232C surtout par le fait qu'il utilise les broches laissées libres pour la boucle de courant 20 mA. Il s'agit des broches :

- 18, recevant les 20 mA de la boucle ;
- 25, fil de retour des 20 mA ;
- 11, émettant les 20 mA ;
- 9, retour d'émission des 20 mA.

Le synoptique de l'interface de la boucle de courant est donné figure 17. Un cavalier permet de passer du mode *boucle de courant* au mode *liaison en tension* ; dans ce dernier cas, la tension différentielle maximale est de 30 V (de + 15 p − 15 V), avec un minimum de 6 V (de + 3 à − 3 V).

L'aspect du connecteur mâle est indiqué figure 18 et ses lignes figure 19. Rappelez-vous qu'on suppose, ici, une connexion à un modem ; par conséquent, et pour une imprimante, il faudrait croiser certains fils (en « modem nul »), ce qu'on va examiner.

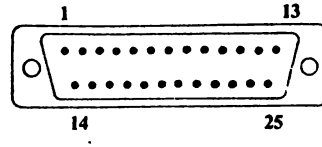


Fig. 18 — Le connecteur mâle 25 broches, type DB, à l'arrière de l'unité centrale pour liaisons séries asynchrones.

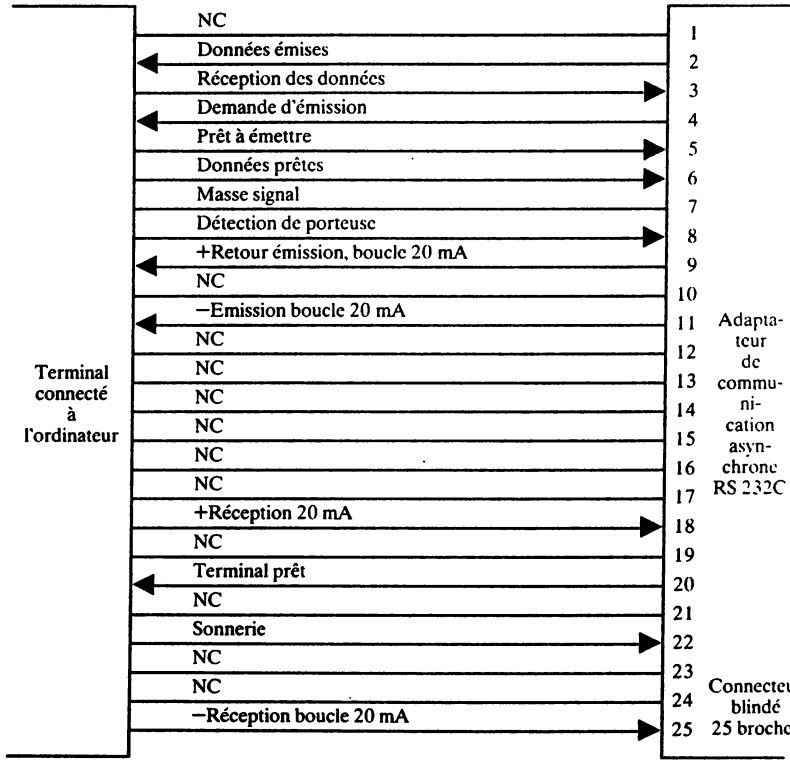


Fig. 19 — La liaison asynchrone entre le micro-ordinateur et le terminal. La marque NC signifie que le fil est « non connecté », inutilisé.

9. USAGE D'UNE IMPRIMANTE SÉRIE AVEC L'IBM-PC

IMPRIMANTE SERIE AVEC IBM-PC

L'usage veut qu'on connecte une imprimante parallèle à l'IBM-PC, réservant la sortie série à un modem. Mais rien n'interdit de brancher une imprimante série via le port série. Il faudra toutefois, et sous système d'exploitation :

- initialiser le protocole d'échange série avec l'imprimante ;
- dérouter les sorties imprimantes vers le port série.

La commande du système d'exploitation (DOS) utilisée à cet effet est MODE.

La première commande, MODE, destinée à la sortie série appelée COM1, sert à définir :

1) La vitesse en bauds : 110, 150, 300, 600, 1200, 2400, 4800 ou 9600.

Il suffit de fournir les 2 premiers chiffres (et pas les zéros).

2) La parité avec (la parité *paire* par défaut) avec :

E = paire
O = impaire
N = pas de parité

3) Le nombre de bits de données, de 7 ou 8 (avec 7, valeur par défaut).

4) Le nombre de bits de stop : 1 ou 2.

5) Dernière indication : un P indique au DOS qu'il s'agit d'une imprimante.

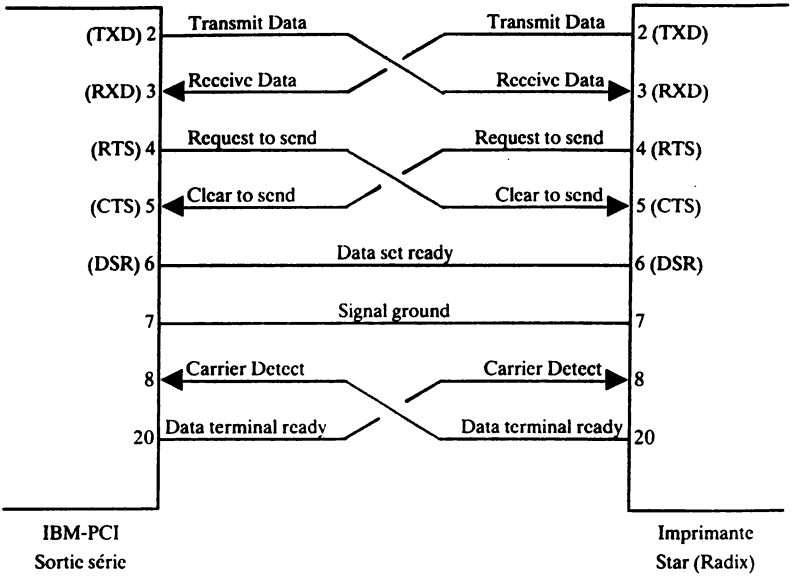
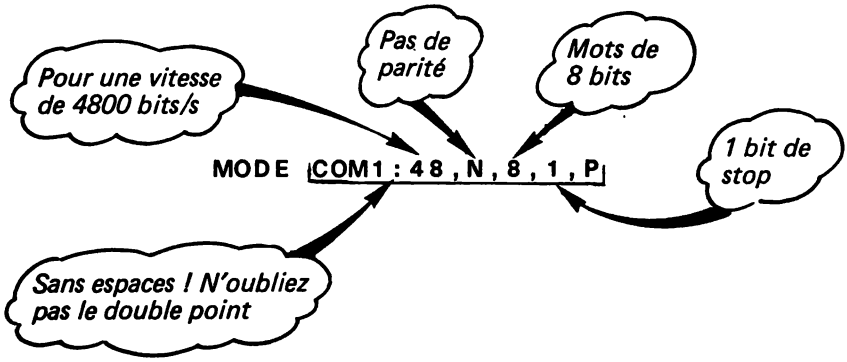
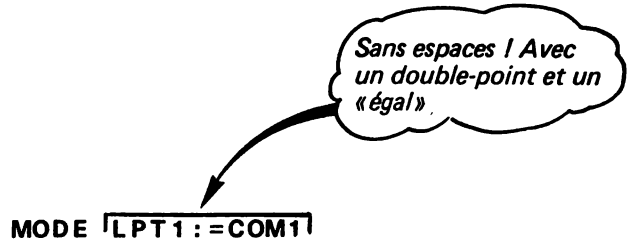


Fig. 20 — Connexion en mode série d'une imprimante à un IBM-PC, ici l'imprimante Radix, de Star. Notez surtout les croisements de fils et comparez au schéma de la figure 6 où nous expliquons le pourquoi de la chose. Nous avons conservé les définitions anglaises EIA, ici, afin que l'on puisse retrouver aisément les sigles correspondants.

L'ordre complet pourra ainsi être le suivant :


Pour une vitesse de 4800 bits/s
Pas de parité
Mots de 8 bits
1 bit de stop
MODE COM1:48,N,8,1,P
Sans espaces ! N'oubliez pas le double point

On y ajoutera la redirection de l'imprimante, appelée LPT1 (« line printer 1 ») sur la sortie série COM1 :



Les deux lignes de code, qu'on pourra introduire dans un fichier AUTOEXEC.BAT sont donc :

```
MODE  COM1:48,N,8,1,P
MODE  LPT1:=COM1
```

Le câble de liaison série ordinateur à imprimante pourra être organisé comme le montre la figure 20 (pour une imprimante *Star/Hengstler*). En fait, la connexion réelle dépendra de l'imprimante ; aussi, ne retenez de ce dessin que l'*obligatoire inversion de fils* puisque la sortie série ne se fait pas sur un modem.

10. LES LIAISONS SÉRIE DU MACINTOSH

MACINTOSH : LIAISONS SERIE

Le micro-ordinateur Macintosh d'*Apple* dispose d'une liaison série générale, type RS-422 pour la liaison, mais avec sortie sur connecteur DB-9, à 9 broches, décrite dans le tableau de la page suivante.

La broche 7 sert à l'échange « en poignée de main » (HSK = « handshake »), bien connu des spécialistes des microprocesseurs, et correspond aux lignes CTS ou TRxC du circuit intégré 8530 de *Zilog*, selon son mode. Dans le cas d'une liaison à imprimante, ce sera l'entrée « Ready » par laquelle l'imprimante indique qu'elle est prête. Dans ce cas, la liaison est dessinée dans la figure 21, pour l'imprimante Imagewriter d'*Apple*. Nous avons pu, toutefois, constater des variantes dans ces liaisons, en particulier avec les Imagewriter 10" et 15".

SORTIE SÉRIE DU MACINTOSH SUR CONNECTEUR DB-9

Broches	Nom		Fonction
	RS-422	RS-232C	
1	Masse	Masse châssis	
2	+5 V		
3	Masse	Masse signal	
4	TxD+		Emission
5	TxD−	TD	Emission
6	+12 V		
7	HSK		Gestion de l'échange
8	RxD+		Réception
9	RxD−	RD	Réception

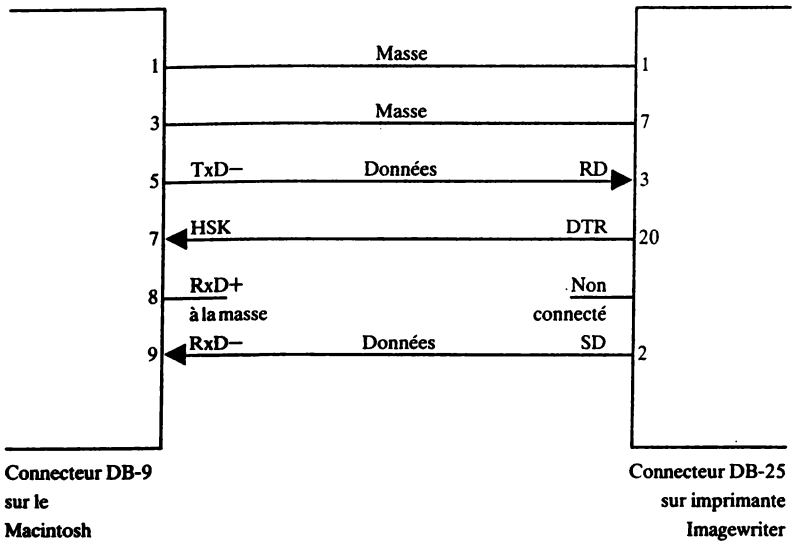


Fig. 21 — Connexion Macintosh à imprimante Imagewriter en mode série, sur connecteur DB-9.

MACINTOSH ET IMPRIMANTE

En fait, une liaison spécifique imprimante est disponible sur connecteur normalisé DB-25, à 25 broches, cette fois. Les lignes employées dépendent du type de l'imprimante ; pour les deux imprimantes principales *Apple*, l'*Imagewriter* et la *Laserwriter*, elles sont indiquées dans le tableau suivant.

LIAISON SÉRIE MACINTOSH A IMPRIMANTE SUR CONNECTEUR DB-25

Broche	Nom	Fonction	Utilisable avec	
			Imagewriter	Laserwriter
1	Masse	—	*	*
2	TXD –	Emission (sortie imprimante)	*	*
3	RXD –	Réception (imprimante)	*	*
4	RTS –	Sortie imprimante	*	*
5	CTS	Entrée imprimante		*
6	DSR	Entrée imprimante		*
7	Masse	—	*	*
8	DCD	Entrée imprimante		*
14	Erreur	En sortie imprimante (désélection)	*	
20	DTR –	Sortie imprimante	*	*
22	Sonnerie	Entrée imprimante		*
Autres	—	Non connectés		

CHAPITRE V

COMMUNICATIONS SYNCHRONES SÉRIE

Toujours en mode série, la liaison synchrone permet des liaisons bien plus rapides que l'asynchrone. On l'appelle encore « mode message ». On examinera aussi dans ce chapitre le cas du mode isochrone.

Les mots-clés de ce chapitre

Liaison synchrones : avantages et inconvénients, 96
Mode message : pour liaison synchrone, 97
Caractère de synchronisation : en mode synchrone, 97

Isochrone : l'un des modes synchrones, 98
Circuit 6852 : interface synchrone, 99
USART : récepteur-émetteur universel synchrone-asynchrone, 104

Circuit 8251 : un USART, 106
Interface synchrone sur IBM-PC : avec le protocole SDLC, 108
Circuit 8273 : contrôleur SDLC, 109

1. AVANTAGES ET INCONVÉNIENTS

LIAISONS SYNCHRONES

On a vu que les liaisons du type *série* pouvaient se faire en *asynchrone* : chaque mot est émis au fur et à mesure de sa disponibilité, et ce quel que soit l'intervalle pouvant séparer deux mots consécutifs.

En synchrone, les mots doivent se suivre, l'un après l'autre, sans espace. Si un mot est de 8 bits, une horloge comptera les octets et indiquera le départ *obligatoire* du mot suivant.

S'il est plus astreignant, ce mode est aussi plus rapide. En effet, en *asynchrone* chaque mot de 1 octet doit être accompagné de bits marquant son début (START) et sa fin (STOP), le *stop* comportant même souvent 1 1/2 ou 2 bits. De ce fait, un octet « utile » se traduit par l'émission d'au moins 10 bits ; il en résulte un « gaspillage » de 20 %. Pour l'éviter, on passe au mode *synchrone* qui exclut une telle perte. Par contre et en synchrone, il faudra fournir une référence de temps au récepteur, une horloge, afin que ce dernier repère bien les octets ; on procède alors selon l'une des deux façons suivantes :

1) — Un fil séparé transmet le signal d'horloge ;

2) — Les informations de temps sont incorporées dans le processus de modulation par le modem.

C'est ce phénomène de *synchronisation* entre l'émetteur et le récepteur qui est à l'origine du nom de *transmission synchrone*, encore appelé *mode message* car une certaine quantité d'octets d'information est transmise à la suite, formant un « message » complet.

En synchrone, le débit peut ainsi excéder très largement 9600 bits par seconde alors qu'en asynchrone, le maximum reste généralement bien en-deçà.

2. LE SIGNAL DE SYNCHRONISATION

MODE MESSAGE

Le *message* transmis peut représenter quelque milliers de caractères (mots). Son émission est précédée par celle d'un *code de synchronisation* destiné au récepteur ; ce code peut être envoyé d'ailleurs non seulement avant chaque message mais encore systématiquement lorsqu'il n'y a pas de message.

CARACTERE DE SYNCHRONISATION

Afin qu'il ne subsiste aucune ambiguïté, le code de synchronisation, ou *caractère de synchronisation* doit être choisi de façon à ne pouvoir être confondu avec un caractère utile, ou une association aléatoire de bits ; en particulier, il ne doit pas présenter de caractère régulier (tel que 01010101). Avec le code ASCII de 1968, c'est le mot 10010110 qui a été retenu, soit 96 en hexadécimal ou 226 en octal. La figure 1 montre ainsi le départ d'un message précédé du caractère de synchronisation.

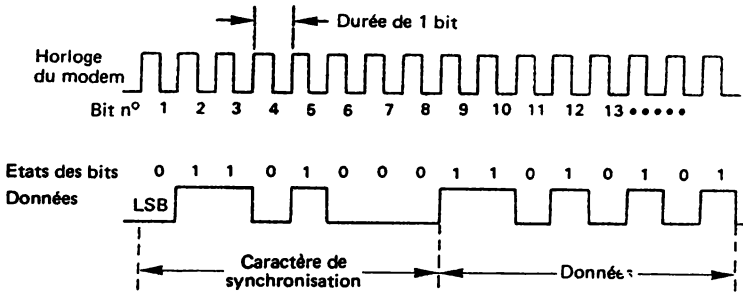

Durée de 1 bit
Horloge du modem
Bit n° 1 2 3 4 5 6 7 8 9 10 11 12 13 • • • • •
Etats des bits 0 1 1 0 1 0 0 0 1 1 0 1 0 1 0 1
Données
LSB
Caractère de synchronisation
Données

Fig. 1 — Chaque message est précédé d'au moins un caractère de synchronisation.

Dans l'attente d'un message, le récepteur surveille ainsi l'arrivée de ce code. Il procède par matériel ou par logiciel. Par exemple, chaque bit reçu est placé dans un registre à décalage dont le contenu est comparé à chaque fois au mot-clé stocké dans un autre registre. En l'absence d'identité, le processus se poursuit jusqu'à la détection d'équivalence des deux mots. A ce moment, le récepteur brandit un *indicateur d'état* (« flag »), cesse sa veille et passe en réception de message. Il comptera les octets successifs comme autant de caractères à la suite, positionnant un indicateur à chaque nouveau caractère.

Pour être cependant certain du départ d'un message, de nombreux équipements de réception identifient *deux caractères de synchronisation successifs,* qui doivent par conséquent être émis. Ce n'est qu'alors qu'ils positionnent l'indicateur de début de message et acceptent celui-ci.

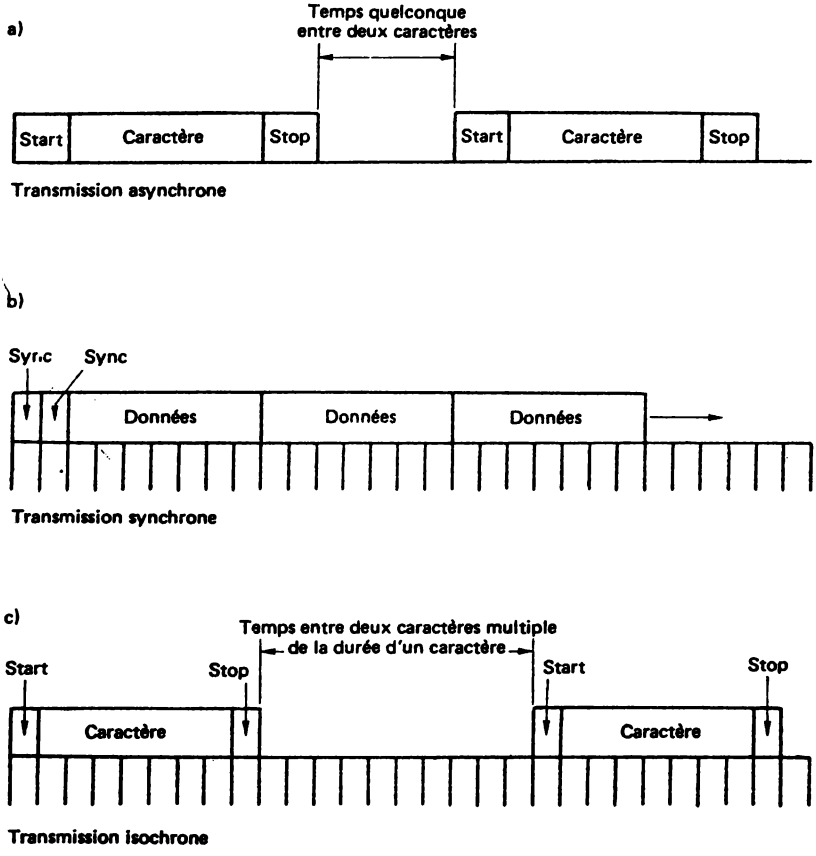


Fig. 2 — Différences entre les modes asynchrone, synchrone et isochrone (ou « isosynchrone »).

Enfin, on peut supposer qu'un risque subsiste pour qu'un caractère de synchronisation soit mal transmis (bruit sur la ligne, parasites...). Dans ce cas, la sécurité voudra que l'émetteur en fournisse non pas deux, mais trois. Pour une meilleure garantie, l'émetteur pourra même en placer quatre sur la ligne ; ainsi, si l'un d'eux est troublé en cours de route, il est certain qu'il en subsistera toujours deux consécutifs « propres à la consommation ». Certaines liaisons font même appel à cinq caractères de synchronisation, afin de réserver une marge de temps suffisante au récepteur pour qu'il se commute en réception après avoir détecté (« vu ») les caractères de synchronisation. Comme on peut le constater, les procédures peuvent ainsi varier largement.

3. LE MODE ISOCHRONE

Le mode de liaison *isochrone* résulte d'un compromis entre le synchrone et l'asynchrone. Comme en asynchrone, chaque caractère est marqué par un bit de start et un bit de stop ; comme en synchrone, l'émetteur et le récepteur doivent être synchronisés. Ce qui est également important, c'est que *l'espace entre deux caractères successifs doit être un multiple exact de la durée d'un bit.* La figure 2 montre ce mode, comparé aux deux autres.

MODE ISOCHRONE

En isochrone, la synchronisation est fournie par l'émetteur. La vitesse de transmission peut atteindre 9600 bits par seconde et n'est dépassée qu'en mode synchrone.

4. UN CIRCUIT D'INTERFACE SYNCHRONE TYPE

CIRCUIT 6852

Divers circuits intégrés monolithiques servant d'interfaces spécialisées pour liaisons synchrones séries ont été commercialisés. Parmi ceux-ci figure le MC6852 de *Motorola,* ou SSDA pour « *Synchronous Serial Data Adapter* ». Il s'insère dans un montage microprocessorisé comme le montre la figure 3, où le microprocesseur est un classique 6800.

L'interface 6852 est adressée par le microprocesseur et apparaît comme deux positions de mémoire. Elle comporte 7 registres internes dont 2, *lus uniquement* pour le 6800, sont le *registre d'états* et le *registre de réception* (qui contient les informations reçues), et 5 registres *à écriture uniquement* dont 3 *registres de commande*, un de *synchronisation* et un *registre d'émission.*

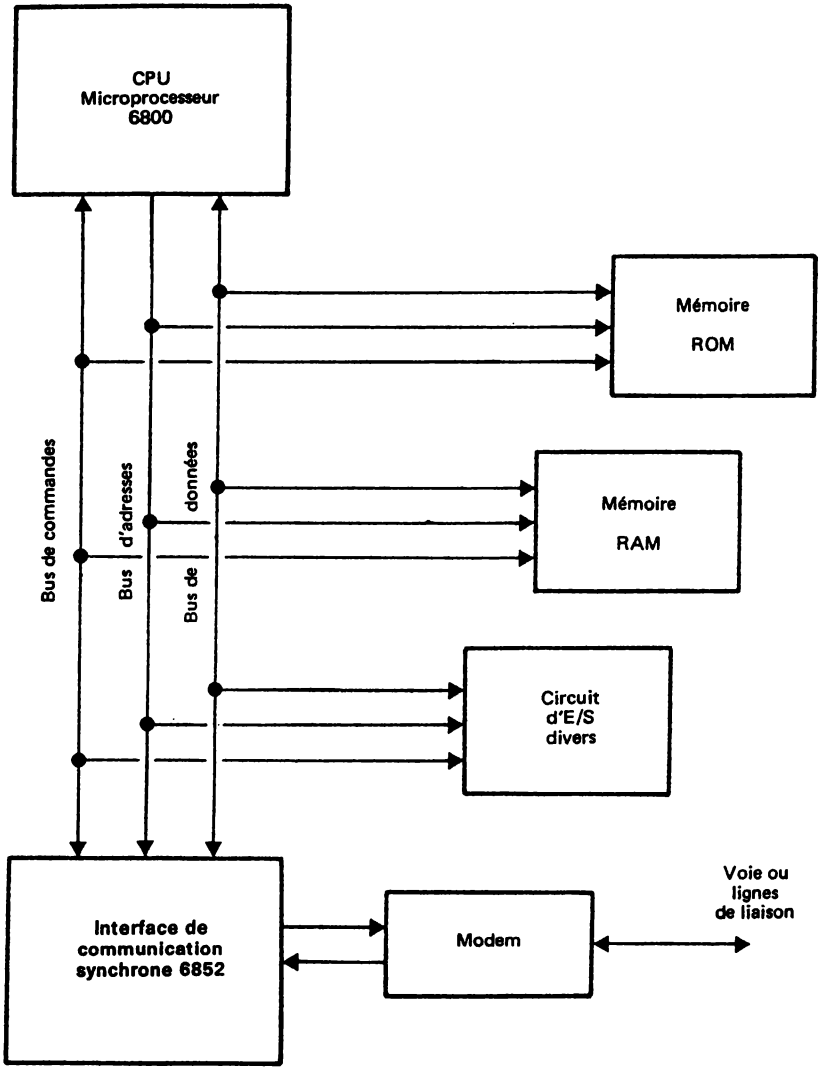


Fig. 3 — Comment l'interface synchrone s'insère dans un montage à microprocesseur.

Les informations à émettre sont chargées dans une pile du type FIFO (« *First-in, First-out* » soit : *premier entré, premier sorti*) qui constitue une file d'attente de 3 mots. Le *registre d'états* témoigne de l'état de cette pile. Les

informations passent ensuite automatiquement de la FIFO au registre à décalage d'émission qui exécute la conversion parallèle à série. Si la FIFO est vide, ce registre reçoit automatiquement un caractère de synchronisation, ou encore des 1 partout. A l'émission, la parité peut être ajoutée à volonté, par programmation (paire, impaire, ou pas de parité du tout). Une commande externe « Clear-to-Send » (CTS) sert à inhiber l'émetteur sans vider la FIFO.

A la réception et en mode de *synchronisation externe,* le récepteur est synchronisé par le signal « Data Carrier Detect » (DCD), de *détection de porteuse ;* il exécute la conversion série-parallèle et transfère les octets dans la FIFO de réception. Le mot de programmation chargé dans l'un des registres de commande précise si la synchronisation se fait sur un ou deux caractères ; pour un caractère, il doit y avoir identité entre le caractère de synchronisation reçu et le mot chargé dans le *registre de synchronisation* avant que les données soient transférées à la FIFO. Si la synchronisation se fait sur deux caractères consécutifs, cette séquence (comparée au contenu du même registre) doit se manifester pour déclencher la réception. La parité est testée et une indication éventuelle d'erreur fournie sous forme d'un indicateur, positionné dans le registre d'états.

La figure 4 montre un synoptique très simplifié de ce circuit et la figure 5, son synoptique détaillé. Le récepteur et l'émetteur disposent chacun de leur horloge ce qui permet leur fonctionnement simultané avec synchronisation séparée. La ligne de demande d'interruption va vers le microprocesseur.

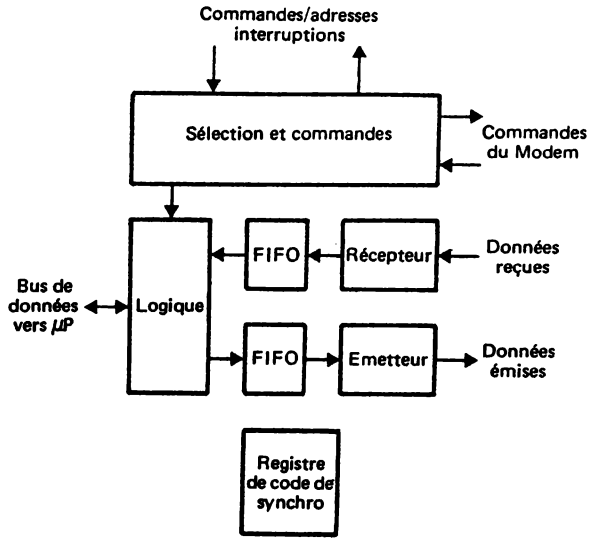


Fig. 4 — Synoptique simplifié du 6852.

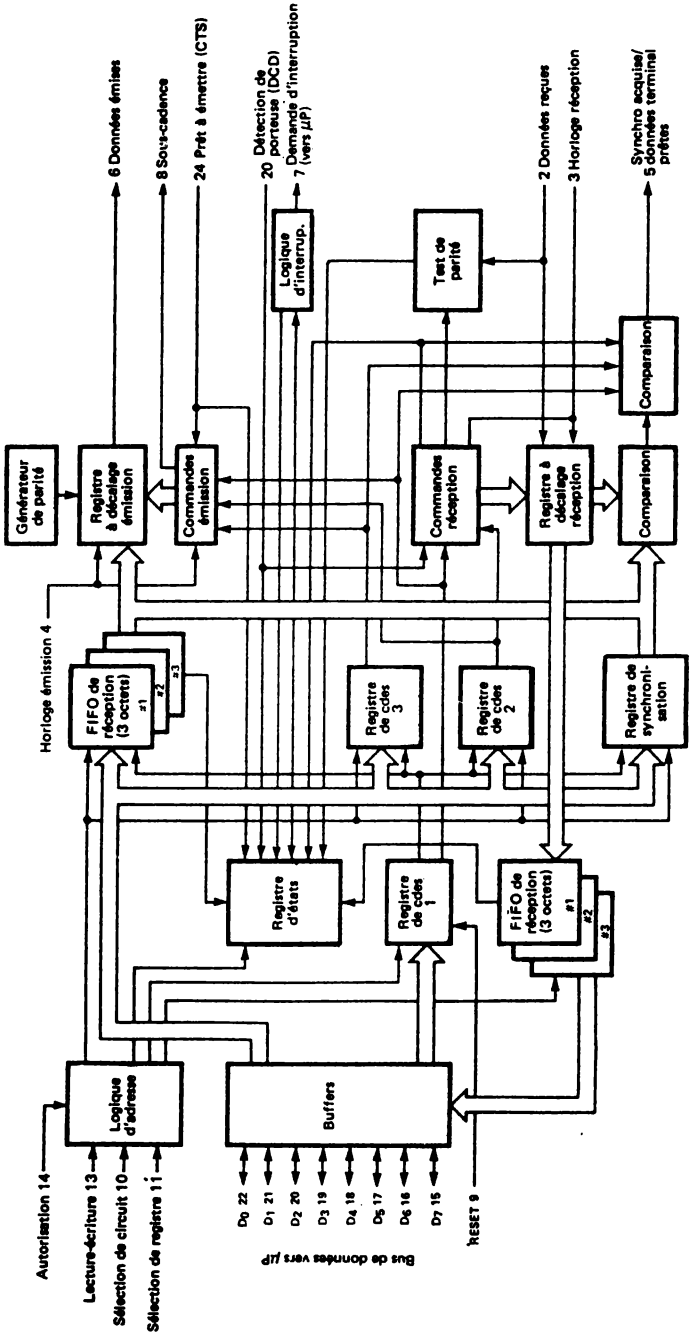


Fig. 5 — Synoptique détaillé de l'interface synchrone 6852.

Initialisation.— A la mise sous tension, le circuit est remis à zéro et doit alors être programmé par le microprocesseur qui charge les mots utiles dans les registres de *code de synchronisation* (code retenu pour le caractère de synchro) et de *commande* (selon les codes proposés et imposés par le fabricant du circuit).

Emission.— Elle est assurée comme décrit ci-dessus. Si, en cours d'émission, le registre à décalage et la FIFO sont vides, des caractères de substitution sont insérés dans la chaîne émise afin de conserver la synchronisation ; ce seront soit un niveau *haut* continu (des 1 partout), soit le caractère de synchronisation contenu dans le registre de cette synchronisation, selon l'état d'un bit de commande programmée. Cet état s'appelle *sous-cadence* (« Underflow ») et il est annoncé par l'envoi d'une impulsion sur la ligne « sous-cadence ».

La ligne *Prêt à émettre,* ou « Clear-to-Send » (CTS) assure une commande automatique de l'émetteur par un dispositif externe, un modem par exemple. Ce dernier émet un signal CTS qui remet à zéro et inhibe l'émetteur, sans agir sur la FIFO d'émission.

Réception.— Puisque les caractères reçus se suivent en série, la synchronisation doit être assurée dès le départ. C'est le rôle du caractère de signalisation, qui sera unique ou doublé selon les cas. Les disques souples et les cartouches magnétiques exigent 16 bits du « préambule » alors que les cassettes se contentent de 8, ce qui décide de l'utilisation de deux ou un mot. Des mots de synchronisation supplémentaires en préambule ou dans le flot des données reçues seront éliminées.

Le récepteur détecte d'eventuelles erreurs de parité qui déclencheront une *demande d'interruption* si les interruptions ont été au préalable autorisées. Une autre cause d'erreur, aboutissant elle aussi à une demande d'interruption, réside dans une vitesse de transmission excessive, une sur-cadence (« Overrun »), qui ne laisse plus au circuit le temps d'acquérir les caractères.

Commande de modem.— Le circuit peut commander un modem ou un périphérique grâce à ses lignes spéciales qui sont : « Clear-to-Send » (CTS), par lequel le modem informe l'interface qu'il est prêt à lui envoyer des caractères ; la *détection de porteuse* qui précède l'envoi d'un caractère de synchronisation initialisant le récepteur ; la *détection de la synchronisation* (« Sync Match ») qui envoie une impulsion de 1 bit sur sa ligne lorsque le code de synchronisation a été détecté ; la même ligne témoigne également de la *disponibilité des données du terminal* (« Data Terminal Ready ») ; enfin, la ligne de *sous-cadence* (« Transmitter Underflow », ou *TUF*), indique qu'un caractère « de remplissage » a été inséré dans le flot.

Performances.— Le 6852 traite des données de 0 à 600 kilobits par seconde. Présenté en boîtier à double rangée de connexions de 24 broches, il consomme typiquement 300 mW sous 5 V. Son horloge peut aller jusqu'à 2 MHz.

5. L'USART

USART

USART est le sigle utilisé pour « *Universal Synchronous-Asynchronous Receiver-Transmitter* », soit *récepteur-émetteur universel synchrone et asynchrone.* C'est un unique circuit intégré qui regroupe les interfaces synchrones et asynchrones, formule à la fois intéressante et économique et dont il existe de nombreuses versions sur le marché, très largement utilisées en micro-informatique.

Un USART type est ainsi le 8251 A de *Intel,* qui s'applique à la famille de microprocesseur 8085 et 8088 ; il intervient dans un système comme le montrent les figures 6 à 9 selon qu'on travaille en asynchrone ou synchrone, avec ou sans modem ; le microprocesseur n'a pas été représenté. Son mode de fonctionnement est déterminé par programmation, via le microprocesseur qui le pilote et doit l'initialiser. Les ordres portent sur *le débit* (de 0 à 64 kilobauds), *la longueur du caractère* (de 5 à 8 bits), *le nombre de bits de stop* (1, 1,1/2 ou 2), *le mode* (synchrone ou asynchrone), *la parité* (paire ou impaire), etc. Le circuit fonctionne en duplex et, pour simplifier, on peut dire qu'il regroupe toutes les caractéristiques des interfaces asynchrones et synchrones.

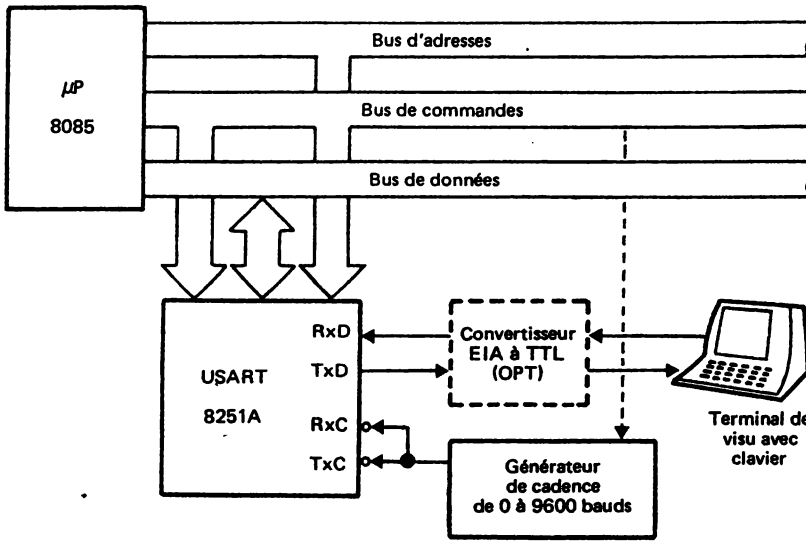


Fig. 6 — Introduction de l'interface universelle dans une liaison série asynchrone. Le générateur de cadence divise l'horloge par 1, ou 16, ou 64 pour fournir le débit, en bauds. Un convertisseur optionnel de niveaux a été prévu.

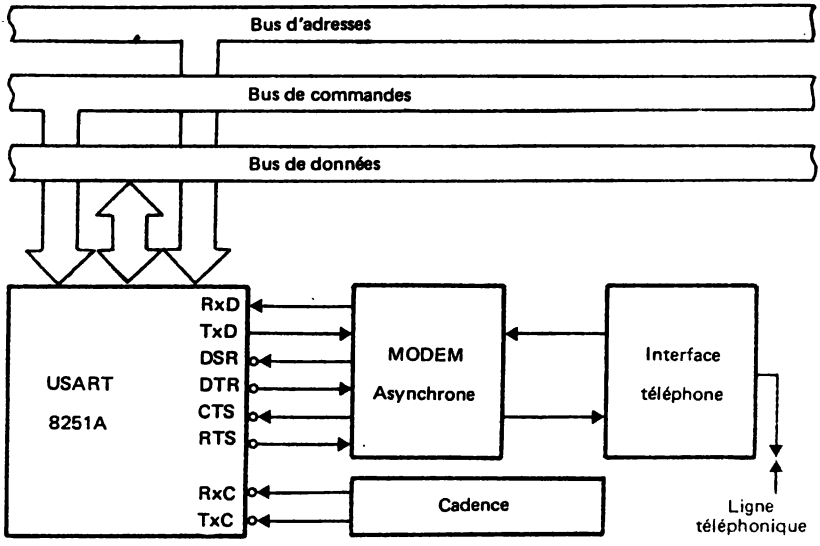


Fig. 7 — L'USART dans une liaison asynchrone avec un modem.

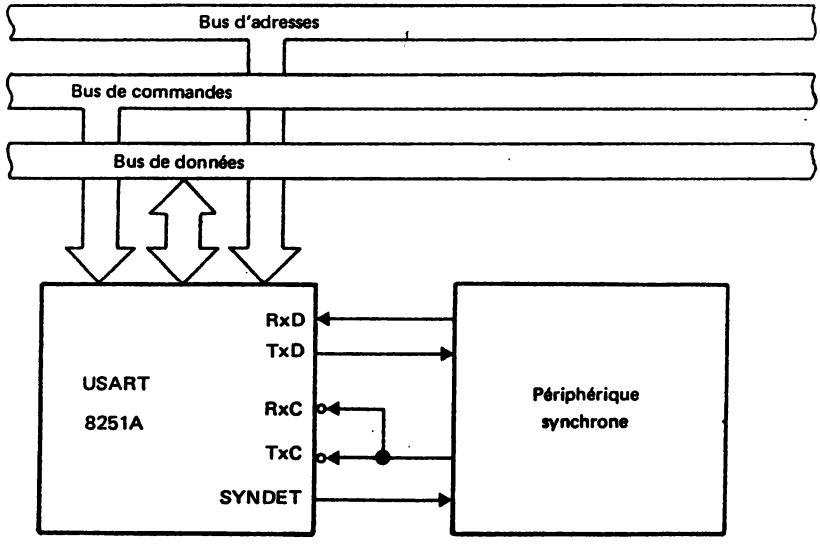


Fig. 8 — L'USART dans une liaison synchrone avec un périphérique.

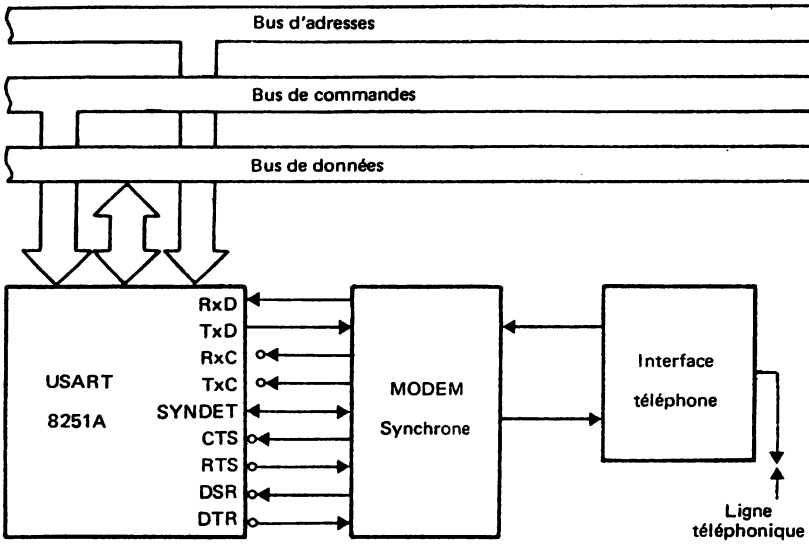


Fig. 9 — L'USART type 8251 A dans une liaison synchrone avec un modem.

CIRCUIT 8251

La structure du 8251 est forcément complexe, le synoptique de la figure 10 n'en donnant qu'une version très simplifiée. En mode *asynchrone,* la transmission de chaque caractère occupe 10 bits alors qu'en *synchrone,* la transmission de *N* caractères demande *N* octets plus 16 bits ; ainsi et au-delà de 8 caractères, le mode synchrone est plus efficace. *On notera que le 8251 peut également fonctionner en isochrone,* mode hybride synchrone-asynchrone.

En synchrone, les caractères peuvent avoir une longueur (programmée) de 5 bits, 6, 7 ou 8 bits, plus une parité optionnelle. La synchronisation est *externe,* via du matériel supplémentaire, ou *interne* grâce à la détection du *caractère de synchronisation,* simple ou doublé.

En mode asynchrone, des bits de stop sont ajoutés et l'horloge peut être de 16 ou 64 fois le débit, en bauds, selon la programmation proposée. Par exemple et pour 110 bauds, l'horloge doit être de 110 Hz, mais aussi de 1,76 kHz pour une division par 16, ou de 7,04 kHz pour une division par 64. C'est ce qui explique la présence d'un circuit externe de cadence, sur les schémas (c'est le « générateur de bauds »). A 2400 bauds, la fréquence sera donc soit de

2400 kHz, soit de 38,4 kHz, soit encore de 153,6 kHz. Si les débits à la réception et à l'émission sont identiques, les deux entrées d'horloge, émission et réception, sont reliées entre elles. En mode synchrone, par contre, le débit est égal à la fréquence d'horloge, sans division.

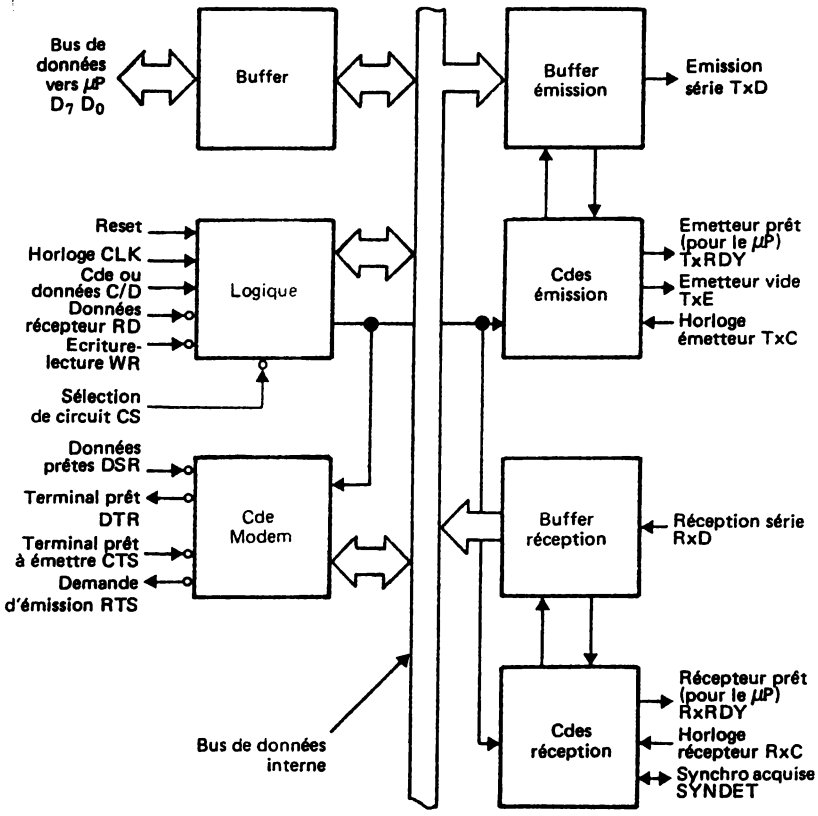


Fig. 10 — Synoptique simplifié du 8251A.

Le mode de liaison est semi-duplex ou duplex. Diverses lignes servent à interfacer un modem, mais on notera que la compatibilité avec les normes RS-232 C de l'IEA n'est pas complète, tant en signaux qu'en tensions ; c'est la raison pour laquelle une interface supplémentaire de conversion de niveaux EIA et TTL a été incluse dans l'un des schémas précédents.

6. L'INTERFACE SYNCHRONE DE L'IBM-PC

INTERFACE SYNCHRONE IBM-PC

Le PC d'IBM peut être doté d'une interface synchrone appliquant le protocole SDLC (« *Synchronous Data Link Control* ») de la Compagnie ; nous allons vous présenter cette interface sans pour autant traiter du SDLC, dont l'étude relève des procédures de communication.

Cette interface fait appel à une carte enfichée dans l'un des connecteurs d'extension 62 broches de la machine et sert à attaquer un modem, en appliquant la norme de liaison RS-232C, avec son connecteur 25 broches, type D. Rappelons, à ce propos, que le RS-232C s'applique aussi bien aux communications asynchrones que synchrones.

Un circuit intégré essentiel de cette carte est le contrôleur SDLC type 8273, de *Intel.* Un classique circuit d'interface parallèle 8255 intervient également afin de développer les possibilités (fig. 11). Le synoptique du 8273, mettant ses registres programmables en évidence, est donné figure 12. Les liaisons sont assurées jusqu'à 9600 bits par seconde sur les niveaux standards RS-232C ; le connecteur utilisé est dessiné dans la figure 13. On notera que la ligne 11, *Sélection attente,* ne répond pas à la norme, pas plus que les lignes 18 et 25 applicables aux seuls modems *IBM.* De nombreuses broches restent inutilisées : 9 et 10, avec 12, 13 et 14, ainsi que 16, 19, 21 et 24.

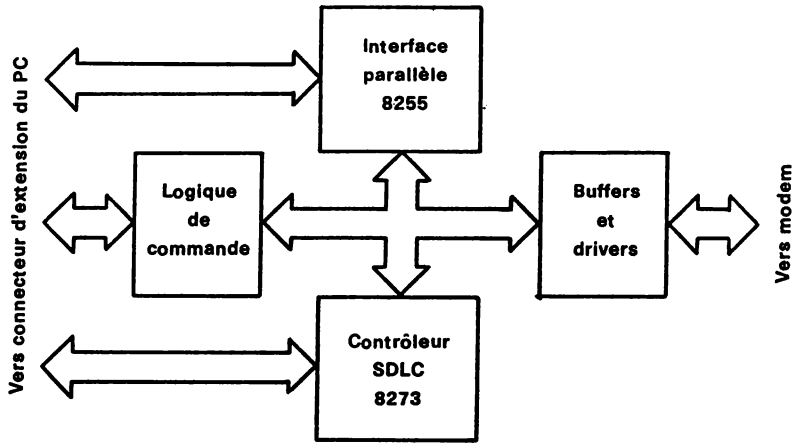


Fig. 11— Synoptique simplifié de la carte d'interface SDLC du PC d'IBM.

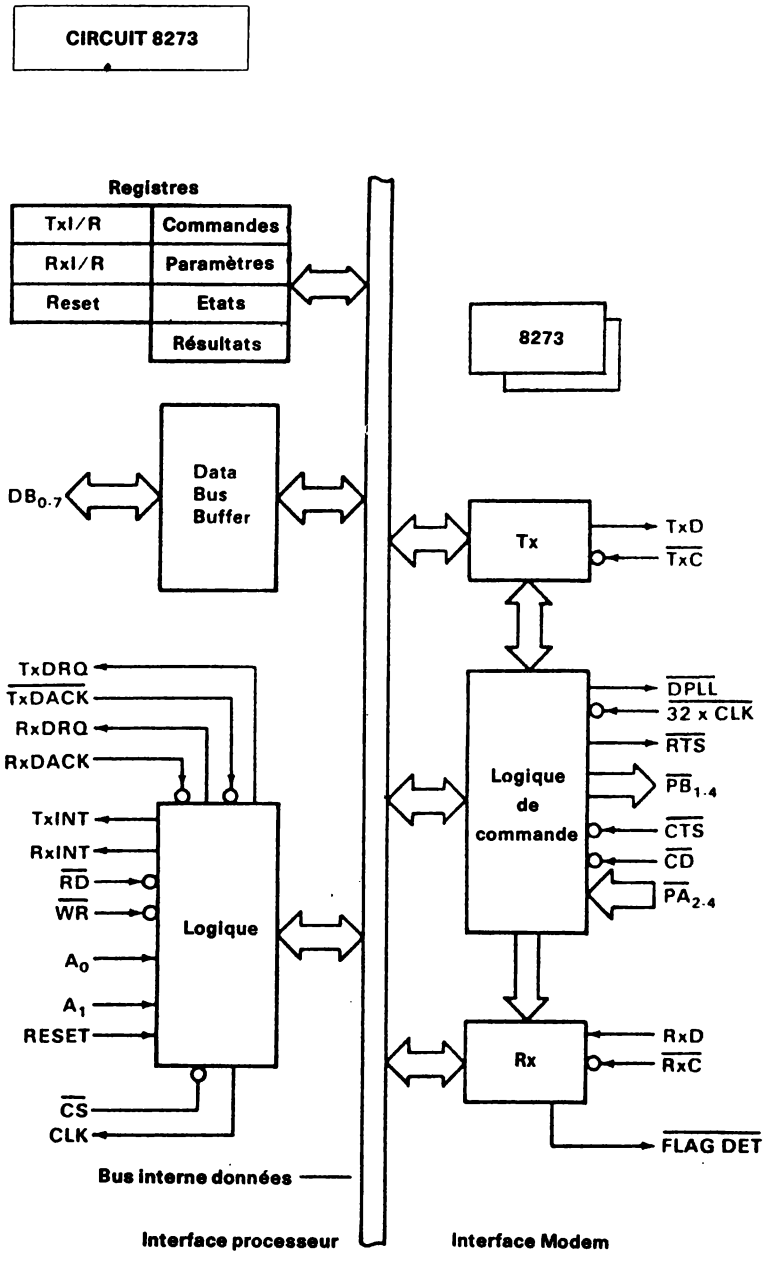
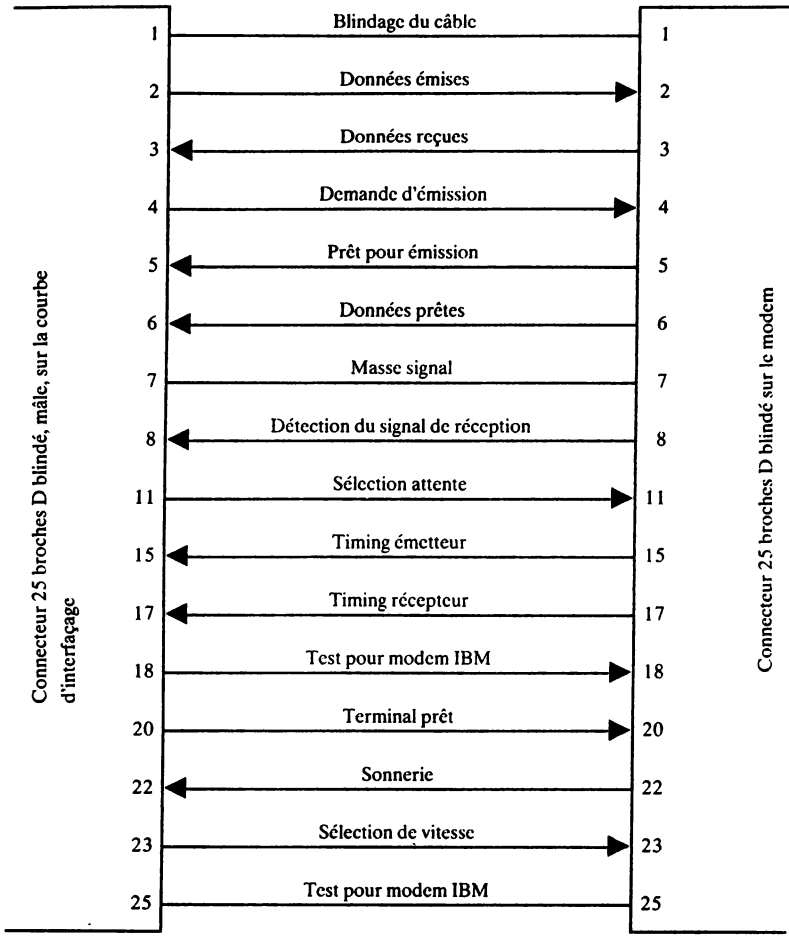


Fig. 12 — Synoptique du contrôleur SDLC type 8273 de *Intel.*

A. La liaison


Connecteur 25 broches D blindé, mâle, sur la courbe d'interfaçage
1 Blindage du câble 1
2 Données émises 2
3 Données reçues 3
4 Demande d'émission 4
5 Prêt pour émission 5
6 Données prêtes 6
7 Masse signal 7
8 Détection du signal de réception 8
11 Sélection attente 11
15 Timing émetteur 15
17 Timing récepteur 17
18 Test pour modem IBM 18
20 Terminal prêt 20
22 Sonnerie 22
23 Sélection de vitesse 23
25 Test pour modem IBM 25
Connecteur 25 broches D blindé sur le modem

B. Le connecteur

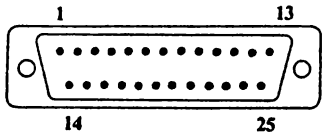


Fig. 13 — Liaison synchrone SDLC à modem sur le PC (en *a*). En *b*, le classique connecteur 25 broches, D, mâle sur la carte d'interface.

CHAPITRE VI

BUS SYSTÈMES ISSUS DES MICROPROCESSEURS

Les fabricants de microprocesseurs ne se sont pas fait faute d'inventer des bus servant à développer leurs systèmes. Certains d'entre eux ont acquis le statut de « bus normalisé ». Voici les plus importants.

Les mots-clés de ce chapitre

Bus d'extension des systèmes, 112
STD : un bus normalisé, 115
Multibus : un bus normalisé, 117
Z-Bus : un bus normalisé, 119

Versabus : un bus normalisé, 119
S-100 : un bus normalisé, 120
VME : un bus normalisé, 123

Future-Bus IEEE-P896 : un bus normalisé, 127
IEEE-P896 : normalisation du Future-Bus, 127
Multibus II : un bus normalisé, 127

1. LES BUS D'EXTENSION

BUS D'EXTENSION DES SYSTEMES

Hormis les entrées-sorties en parallèle et les entrées-sorties en série, il existe un mode de liaison parallèle, appelé *bus* également, qui vise surtout à l'extension des systèmes. Le plus souvent, il s'agira d'un ensemble de lignes donnant accès aux bus internes du système à microprocesseur.

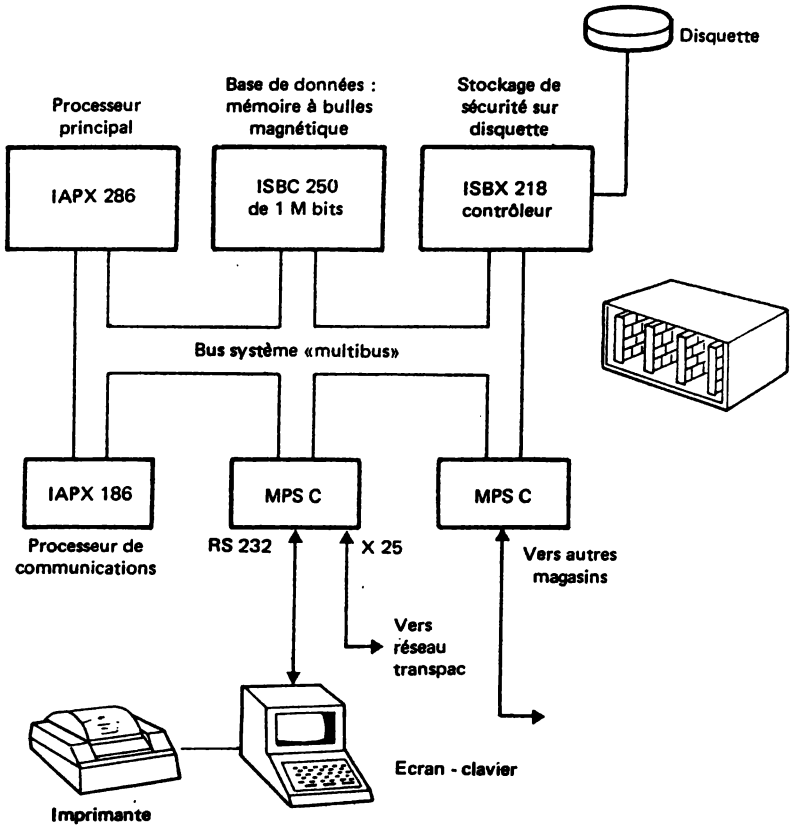


Fig 1 — Système de gestion de stocks montrant l'intervention et le rôle du bus : il relie tous les éléments du système. On remarquera, ici, l'usagenouveau de mémoires à bulles magnétiques pour le stockage principal, et les interconnexions possibles via le réseau Transpac avec d'autres magasins de la chaîne.

La figure 1 montre, à titre d'exemple, comment un « Multibus » (nom de marque du bus *Intel*) assemble un système de gestion de stocks. Les informations peuvent circuler dans tous les sens possibles, ce qui fait qu'on compare souvent un bus à une autoroute desservant de nombreuses entrées-sorties.

De très nombreux bus ont été définis, à peu près autant qu'il y a de microprocesseurs. Quelques-uns d'entre eux ont fait l'objet d'une normalisation et reçu un nom particulier : Multibus, Unibus, STD, S-100, Qbus... L'ennui majeur, c'est qu'ils ne sont absolument pas compatibles entre eux ; d'autre part, on ignore toujours lesquels s'imposeront ou tout au moins survivront, et lesquels disparaîtront après une vie éphémère. Ce que l'on est en droit de supposer, c'est que les grands fabricants de produits sauront de toutes façons imposer les leurs.

Il en résulte qu'actuellement, un matériel conçu pour être relié à un bus d'un type donné ne pourra être connecté à un autre, ce qui peut être très regrettable. Théoriquement il est toujours possible de changer la destination d'une carte ou d'un sous ensemble et de les adapter à un nouveau bus ; mais en pratique, une telle modification implique le plus souvent un énorme travail de reconversion et de création, matérielle et logicielle, et par conséquent, un investissement tel qu'il vaut mieux y renoncer.

Comment définit-on un bus ? Puisqu'il s'agit d'une voie parallèle de liaison, il faut d'abord préciser le nombre de « fils », de voies, qu'il comporte, puis les types de signaux que ceux-ci doivent véhiculer, et enfin les procédures d'échanges. On y ajoutera l'organisation éventuelle des connecteurs mâles et femelles associés, et l'organisation de son câble. Ainsi, une définition complète d'un bus — on dit : *sa normalisation* — porte sur :

1 - L'organisation matérielle ;

2 - Le fonctionnement ;

3 - Les procédures d'échanges.

Pour illustrer ce thème, voici quelques bus courants, dans une présentation très succincte.

LES BUS ET LEURS MICROPROCESSEURS

Bus	Société	Micro-processeur	Adressage sur	Notes
Versabus . .	Motorola	68000	32 bits	Pour μP 8, 16 et 32 bits
Multibus . . IEEE-795	Intel	8080 8085 8086...	20 bits	Pour μP 8 et 16 bits
Qbus	DEC	LSI-11	18 bits	Pour μP 16 bits
Z-bus	Zilog	Z8000	32 bits	Pour μP 8, 16 et 32 bits
STD IEEE-696	Prolog	8080, 8085 6800, 6809 Z80 ...	16 bits	Pour μP 8 bits
STD-Z80 . .	Mostek	Z80	16 bits	Pour le Z80
S-100	MITS	8080	16 bits	Pour l'informatique «personnelle»
IEEE-488 .	(HP)	Indifférent	5 bits	Pour l'instrumentation

2. BUS STD

LE BUS STD

Mis au point par *Prolog*, le bus « Standard », ou STD, est destiné à plusieurs microprocesseurs 8 bits tels que les 8080, 8085, Z80, 6800, 6809... et constitue, en ce sens, une tentative intéressante de normalisation réelle multi-famille. Il se compose de 56 fils répartis en :

- 6 pour les alimentations ;
- 8 pour les données ;
- 16 pour les adresses ;
- 22 pour les commandes ;
- 4 supplémentaires pour les alimentations de puissance.

Le total fait bien 56, et se connecte sur des cartes en double face de deux fois 28 contacts séparés de 0,125 pouce. La distribution et l'affectation de chaque contact, de chaque broche du connecteur, est rigoureuse et répond à un cahier précis de standardisation. Par exemple, le + 5 V passe sur les fils 1 et 2, et la masse principale sur 3 et 4. L'ensemble des fils de commande, qu'on peut regrouper en sous-bus de commandes, comprend un certain nombre de fonctions répondant aux spécifications générales des microprocesseurs concernés et incluant le traitement des interruptions et les échanges en poignée de mains.

Ainsi, des modules, des cartes ou des sous-ensembles d'origines différentes conçues selon la normalisation STD pourront-elles être assemblées immédiatement, servant à réaliser, par exemple, le système industriel de la figure 2. Notons toutefois que la longueur des lignes est généralement réduite à la longueur de la carte mère.

Lorsque plusieurs périphériques veulent simultanément accéder au bus, la priorité est régie par une chaîne série, chaque module incorporant sa propre logique de priorité. Une structure de priorité parallèle peut également être réalisée.

Par la suite, *Mostek* a défini un sous-ensemble du STD exclusivement destiné au microprocesseur Z80 et appelé STD-Z80 ; il est accompagné d'une définition exacte et fonctionnelle des différents signaux et de leur timing, que ne comprenait pas le STD de *Prolog*, et dispose actuellement d'un jeu important de cartes qui lui sont directement compatibles.

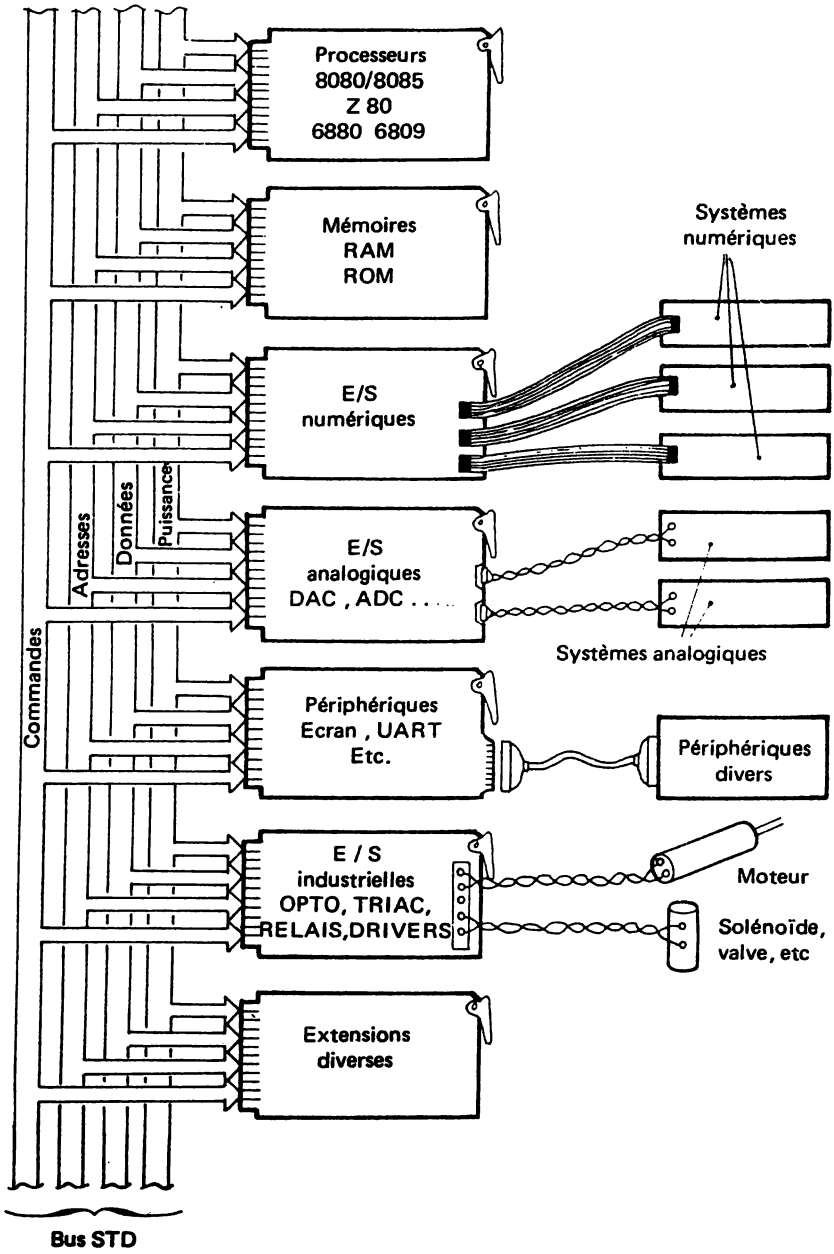


Fig. 2 — Une application du bus « Standard ».

3. LE MULTIBUS

MULTIBUS

Le Multibus de *Intel*, lui, se présente sur 86 fils répartis en :

- 12 pour les alimentations ;
- 12 pour les commandes ;
- 10 pour le contrôle et les adresses ;
- 8 pour les interruptions ;
- 16 pour les adresses ;
- 16 pour les données ;
- 12 supplémentaires pour les alimentations.

Il est destiné aux microprocesseurs *Intel* 8 et 16 bits parmi lesquels les 8080, 8085, 8086... et peut adresser, sur 20 bits, un méga-octet en mémoire, avec extension prévue à 24 bits pour 16 méga-octets.

Conçu selon le concept « maître-esclave », où le « maître », quel qu'il soit, prend la direction des échanges, il peut transférer des informations jusqu'à 5 mégahertz ; en outre, il supporte des systèmes en multitraitement.

Ce bus asynchrone, non multiplexé, a été normalisé sous la référence IEEE-795. Il supporte une longueur de ligne de 15 pouces (environ 40 cm), et 16 modules. La priorité est gérée soit par chaînage, soit en mode parallèle, via un codeur de priorités (type 74148 avec 74 S 138 en décodeur, par exemple).

Ce bus recourt à un connecteur de 86 broches décrit dans le tableau : les broches *impaires* sont du côté composants, les broches *paires* du côté circuit.

BROCHAGE DU MULTIBUS INTEL

Groupe de fonction	Broches	Nom	Description
Alimentations	1 et 2	GND	Masse signal
	3 à 6	+5	5 V
	7 et 8	+12	12 V
	9 et 10	−5	−5 V
	11 et 12	GND	Masse signal
Commandes	13	BCLK	Horloge bus (Bus Clock)
	14	INIT	Initialisation
	15	BPRN	Entrée priorité bus (Bus Priority In)
	16	PPRO	Sortie priorité bus (Bus Priority Out)
	17	BUSY	Bus occupé
	18	BREQ	Demande de bus (Bus request)
	19	MRDC	Lecture mémoire (Memory Read Command)
	20	MWTC	Ecriture mémoire (Memory Write Command)
	21	IORC	Lecture E/S (In/Out Read Command)
	22	IOWC	Ecriture E/S (In/Out Write Command)
	23	XACK	Acquisition XFER (XFER Acknowledge)
	24	INH1	Inhibition 1 RAM
	25	AACK	Acquisition avancée (Advanced Acknowledge)
	26	INH2	Inhibition 2 ROM
	27 à 30	—	Réservé
	31	CCLK	Horloge constante (Constant Clk)
	32 à 34	—	Réservé
Interruptions	35	INT6	Demandes d'interruptions parallèles
	36	INT7	
	37	INT4	
	38	INT5	
	39	INT2	
	40	INT3	
	41	INT0	
	42	INT1	
Adresses	43 à 58	ADR...	16 bits d'adresse
Données	59 à 74	DAT...	16 bits de données
Alimentations	75 et 76	GND	Masse signal
	77 et 78	−10	−10 V
	79 et 80	−12	−12 V
	81 à 84	+5	+5 V
	85 et 86	GND	Masse signal

4. LE Z-BUS

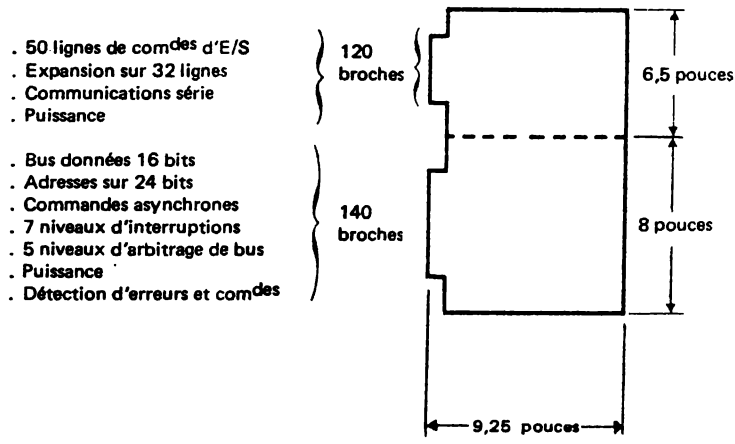
Z-BUS

Créé par *Zilog* pour le microprocesseur 16 bits Z8000, le Z bus est à 96 voies dont 32 pour les adresses et données et 21 pour les alimentations, 9 pour les interruptions, et 28 pour les commandes, le reste étant en réserve. Ce bus est appelé à servir aussi bien le Z8000 que les Z8 et Z80, sur 8 bits, ou que de futurs microprocesseurs 32 bits. Son succès a été tout relatif, à la mesure du succès des Z8000.

5. LE VERSABUS

VERSABUS

Motorola, avec son « Versabus », a mis au point une structure de bus destinée aussi bien aux microprocesseurs 8 et 16 que 32 bits. Au prix, cependant, d'un accroissement du nombre de broches qui atteint 260 pour les deux connecteurs prévus (fig. 3). La vitesse de transfert est de 5 MHz ; toutes les possibilités de système, multitraitement et autres, sont prises en compte. L'adressage, sur 32 bits maximum, atteint 4 milliards de mots.



(1 pouce = 25,4 mm)

Fig. 3 — Organisation de la carte imprimée et du brochage pour le Versabus.

Il s'agit, à nouveau, d'un bus asynchrone, non multiplexé, avec chemin de données sur 8 bits, 16 ou 32 bits, et champ d'adressage sur 32 bits. Les lignes de signal peuvent atteindre 18 pouces (un peu moins de 50 cm) ; 15 cartes peuvent être supportées par un fonds de panier Versabus.

6. LE BUS S-100

BUS S-100

A l'inverse, le S-100 est l'un des plus « anciens » bus normalisés, par hasard, et pour le microprocesseur 8080. Il est typique des « ordinateurs personnels » grand public et a été créé en 1976 par la société *MITS* qui, en imaginant l'ordinateur individuel « Altair », donnait le coup d'envoi à ce marché.

Il est non multiplexé, asynchrone aussi, peut supporter jusqu'à 22 cartes ou périphériques. Les connexions peuvent atteindre 25 pouces (environ 80 cm) pour le signal. Son connecteur de 100 broches se répartit en :

- 8 lignes d'entrées
- 8 lignes de sorties
- 24 lignes d'adresses
- 8 lignes d'état
- 5 de commandes de sorties
- 6 de commandes d'entrées
- 8 de DMA
- 8 d'interruptions vectorisées
- 20 pour des ordres divers

Les autres restent inutilisées. On remarquera que, grâce à son double jeu de lignes de données sur 8 bits, en entrées *et* en sorties, il peut gérer des échanges sur deux octets. Ajoutons qu'il a été normalisé sous le nom de IEEE-696.

Son brochage est indiqué dans le tableau ; les broches 1 à 50 sont situées *côté composants,* 51 à 100 *côté circuit.* On remarquera qu'il ne comporte pas de lignes distinctes de lecture mémoire (MEMR) et d'écriture mémoire (MEMW), ce qui oblige à un décodage ultérieur.

BROCHAGE DU CONNECTEUR S-100

Broches	Nom	Fonction
1	+8 V	8 V continus non régulés
2	+16 V	16 V continus non régulés
3	XRDY	« Ready » externe
4 à 11	V1Ø à V17	Interruptions vectorisées Ø à 7
12 à 17	—	Réserve
18	STATUS DSBL	Inhibition états
19	CCD DSBL	Inhibition commande
20	—	Réserve
21	SS	Pas à pas (Single Step)
22	ADDR DSBL	Inhibition adresse
23	DØ DSBL	Inhibition sortie donnée
24	Φ1	Horloge, phase 1
25	Φ2	Horloge, phase 2
26	PHLDA	Acquisition du « Hold »
27	PWAIT	Attente (Wait) processeur
28	PINTE	Interruption processeur autorisée
29	A5	Bit d'adresse 5
30	A4	4
31	A3	5
32	A15	15
33	A12	12
34	A9	9
35	DO1	Sortie bit de donnée 1
36	DOØ	Sortie bit de donnée Ø
37	A1Ø	Adresse, bit 1Ø
38	DO4	Sortie bit de donnée 4
39	DO5	5
40	DO6	6
41	DI2	Entrée bit de donnée 2
42	DI3	3
43	DI7	7
44	SM1	Etat M1
45	SOUT	Sortie états
46	SINP	Entrée états

(suite au verso)

BROCHAGE DU CONNECTEUR S-100

(suite du tableau de la page précédente)

Broches	Nom	Fonction
47	SMEMR	Lecture état mémoire
48	SHLTA	Etat acquisition du « Hold »
49	CLOCK	Horloge
50	GND	Masse signal
51	+8 V	8 V continus non régulés
52	−16 V	− 16 V continus non régulés
53	—	Réserve
54	EXT CLR	« Clear » externe
55 à 67	—	Réserve
68	MWRITE	Ecriture mémoire
69 et 70	—	Réserve
71	RUN	« Run »
72	PRDY	« Ready »
73	PINT	Interruption processeur autorisée
74	PHOLD	« Hold » processeur
75	PRESET	« Reset », ré-initialisation
76	PSYNC	Synchronisation
77	PWR	Ecriture processeur
78	PDBIN	Entrée bus donnée
79	AØ	Bit d'adresse Ø
80	A1	1
81	A2	2
82	A6	6
83	A7	7
84	A8	8
85	A13	13
86	A14	14
87	A11	11
88	DO2	Sortie bit de donnée 2
89	DO3	3
90	DO7	7
91	DI4	Entrée bit de donnée 4
92	DI5	5
93	DI6	6
94	DI1	1
95	DIØ	0
96	SINTA	Acquisition état interruption
97	SWO	Sortie état écriture
98	SSTACK	Etat pile
99	POC	Initialisation sur mise en service alimentation
100	GND	Masse signal

7. LE BUS VME

BUS VME

Il est rare que de grands fabricants de microprocesseurs s'entendent pour définir, en commun, une norme. Or, c'est bien ce qui s'est passé puisque *Motorola, Mostek* et *Philips/Signetics* se sont mis d'accord, en 1981, pour proposer un nouveau bus standard, le VME (de « *Versa Module Eurocard* »). Ils ont été rejoints par *Thomson-Efcis,* qui a fait sienne cette norme.

En raison de la puissance de ses promoteurs, le bus VME a donc toutes chances d'être bien accueilli, puisqu'il répond aux normes DIN 41612 et 41494, pour circuits imprimés et fonds de panier. Il s'applique aux systèmes à 8, 16 et 32 bits et dérive du Versabus de *Motorola,* utilisé avec le microprocesseur 68000. Parmi ses principales caractéristiques, citons :

- il supporte des systèmes mono et multiprocesseurs ;
- il supporte des architectures 32 bits ;
- il permet le transfert de données à 24 M octets par seconde (48 M maximum) ;
- il incorpore un protocole de transfert asynchrone non multiplexé ;
- il dispose de quatre lignes hiérarchisées de demande de transfert ;
- il supporte des systèmes distribués ou centralisés, avec gestion des interruptions et sept niveaux d'interruptions ;
- il offre un canal de communication série distinct ;
- il inclut des dispositifs accroissant la fiabilité, avec des indicateurs d'erreur de bus, de défaillance système, et de défaillance de l'alimentation ;
- la même carte fond de panier peut supporter jusqu'à 20 charges, sur une longueur de 19 pouces (environ 50 cm).

Ce bus est articulé autour d'un double connecteur, P1 et P2. Le connecteur P1, à 96 contacts, comporte 24 lignes d'adresses (adressage de 4 giga-octets) et 16 lignes de données (fig. 4) ; le second connecteur, P2, étend ces possibilités pour une carte double et porte les bus d'adresses et de données à 32 bits.

La figure 4 montre les dimensions possibles des cartes. La carte Europe de base mesure 100 × 160 mm ; elle est comparable, d'ailleurs, à la carte standard pour bus STD de *Prolog/Mostek*. Le format 160 × 233,68 mm appelé « double Europe » est, lui, comparable à celui du S-100 et du Multibus.

La distribution des signaux sur P1 et P2 est résumée tableau 1, les affectations des broches de P1 étant données dans le tableau suivant avec le nom des signaux et pour les trois rangées de 32 contacts chacune. Tous les signaux nécessaires pour un système 16 bits ont été regroupés sur P1.

Les créateurs du VME l'ont comparé au Multibus de *Intel*, au Q-Bus de *Digital Equipment* et au « Futurebus », ou bus IEEE P 896, encore en projet de normalisation à l'IEEE (tableau). La commercialisation des cartes aux normes VME avait démarré dès la fin 81 avec des cartes unités centrales *Motorola* (à 68000) et des cartes-mémoires.

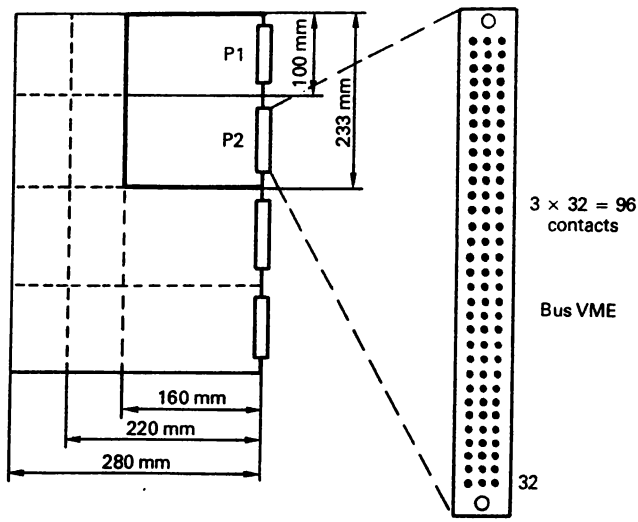


Fig. 4 — Formats : la carte Europe simple mesure 10 × 16 cm, et la double, 16 × 23,3 cm. La longueur des cartes pourrait être étendue pour rejoindre d'autres formats. P1 et P2 sont les deux connecteurs à 96 broches.

BUS VME

Signaux primaires (connecteur P1)		
Bus d'adresses	A01 à A23	Adressage primaire : 16 M mots
Bus de données	D00 à D15	Sur 16 bits
Commandes d'adresses	AS AM0 à AM5	Strobe d'adresse Modificateurs d'adresses
Commandes de données	DS0, DS1 WRITE DTACK LWORD	Strobes Direction lecture/écriture Acquisition de transfert Sur 32 bits
Interruptions	IRQ1-IRQ7 IACK, IACKIN IACKOUT	7 niveaux Acquisition des interruptions et chaînage
Arbitrage de bus	BRD-BR3 BG0-3IN/ BG0-3OUT BBSY, BCLR	Demande de bus (4 niveaux) Chaînage des acquisitions bus Bus occupé, bus libre
Canal série interprocesseurs	SERDAT SERCLK	Données en série Horloge
Signaux d'erreur	SYSFAIL BERR ACFAIL	Défaillance système Erreur bus Défaillance alimentation
Alimentation	+5, ±12 +5 veilleuse GND	Comprend la mise en repos (sauvegarde)
Divers	SYSCLK SYSRESET	Horloge 16 MHz Reset système
Signaux d'extension (connecteur P2) — Bus VME		
Bus adresses	A24 à A31	Adressage de 4 gigabits
Bus données	B16 à D31	Données sur 32 bits
Alimentation	+5 GND	Lignes supplémentaires
Entrées-sorties	Utilisateur	64 broches disponibles

AFFECTATION DES BROCHES DE P1 DU BUS VME

Broche	Rangée		
	A	B	C
1	D_0	BBSY	D_8
2	D_1	BCLR	D_9
3	D_2	ACFAIL	D_{10}
4	D_3	BG0IN	D_{11}
5	D_4	BG0OUT	D_{12}
6	D_5	BG1IN	D_{13}
7	D_6	BG1OUT	D_{14}
8	D_7	DG2IN	D_{15}
9	GND	BG2OUT	GND
10	SYSCLK	BG3IN	SYSFAIL
11	GND	BG3OUT	BERR
12	DS_1	BR_0	SYSRESET
13	DS_0	BR_1	LWORD
14	WRITE	BR_2	AM_5
15	GND	BF_3	A_{23}
16	DTACK	AM_0	A_{22}
17	GND	AM_1	A_{21}
18	AS	AM_2	A_{20}
19	GND	AM_3	A_{19}
20	IACK	GND	A_{18}
21	IACKIN	SERCLK	A_{17}
22	IACKOUT	SERDAT	A_{16}
23	AM_4	GND	A_{15}
24	A_7	IRQ_7	A_{14}
25	A_6	IRQ_6	A_{13}
26	A_5	IRQ_5	A_{12}
27	A_4	IRQ_4	A_{11}
28	A_3	IRQ_3	A_{10}
29	A_2	IRQ_2	A_9
30	A_1	IRQ_1	A_8
31	− 12 V	+ 5 V STDBY	+ 12 V
32	+ 5 V	+ 5 V	+ 5 V

8. LE FUTURE BUS IEEE-P896

FUTURE BUS
IEEE-P896

Proposé sous la normalisation de IEEE-P896, le « Future bus » est un bus synchrone et multiplexé, capable de relier des modules asynchrones. Sa voie de donnée est sur 32 bits, mais aussi sur un ou deux octets, et il dispose, au choix, d'un faisceau de 16 lignes d'adressage, ou 24, ou 32. Il peut supporter 32 cartes sur une distance de 50 cm (fond du panier). Le connecteur 96 broches est un DIN 41612 ; 64 sont employées en mode de base. Nous le comparons, dans le tableau ci-après, au VME, au Multibus et au Q-Bus de *Digital Equipment.*

9. LE MULTIBUS II

MULTIBUS II

Datant de 1983, le Multibus II constitue une extension du Multibus, appelé d'ailleurs Multibus I par la suite. Il en étend la bande passante, les capacités en multitraitement et en performances. En fait, il se compose de cinq bus qui sont :

- le bus système parallèle, ou iPSB (avec *i* pour *Intel,* son concepteur),
- le bus local d'extension, ou iLBX (« local bus extension »),
- le bus système série, ou iSSB,
- deux bus empruntés à l'architecture du Multibus I : l'iSBX, *bus d'expansion des entrées-sorties* et le *bus multicanal d'accès direct à la mémoire.*

QUATRE BUS COMPARÉS

Bus / Caractéristiques	Bus VME	Multibus IEEE/796	Q-Bus	IEEE P 896 « Futurebus »
Type de bus	Asynchrone Non multiplexé	Asynchrone Non multiplexé	Asynchrone Multiplexé	Synchrone : Multiplexé
Largeur du bus d'adresses	24 standard extension à 32	20 standard extension à 24	16 standard extension à 18	32 standard
Largeur du bus de données	8, 16, 32	8, 16	8, 16	8, 16, 32
Taille de la carte	160 × 100 mm 160 × 233,4 mm	171,5 × 304,8 mm	214 × 147,3 mm 214 × 263,4 mm	N'importe quelle carte d'Europe
Connecteurs	Rapportés	Encartables	Encartables	Rapportés
Nombre de broches connecteurs primaires/ secondaire	96/96	86/60	38	64 ou 96
Alimentations	+ 5 V ; ± 12 V + 5 V (repos)	+ 5 V ; ± 12 V	+ 5 V ; ± 12 V (5 V ; + 12 V pour sauvegarde)	+ 5 V

QUATRE BUS COMPARÉS (suite)

Caractéristiques \ Bus	Bus VME	Multibus IEEE/796	Q-Bus	IEEE P 896 « Futurebus »
Niveaux d'interruptions	7	8	4	0 (utilisation du bus série)
Niveaux d'accès au bus	4	1	1	32
Multimicroprocesseur ?	Oui	Oui	Non (maître-esclave)	Oui (avec superviseur)
Signaux de test	Perte d'alimentation, panne système, erreur bus	Non	Perte d'alimentation continue ou secteur	Code autocorrecteur
Cycles spéciaux	Lecture/modification écriture, transfert de bloc niveaux privilégiés d'accès	Verrouillage	Rafraîchissement, lecture/modification/écriture, interruption sur événement	Lecture/modification/écriture transfert de bloc cycles divers
Bus série séparé	Oui	Non	Non	Oui

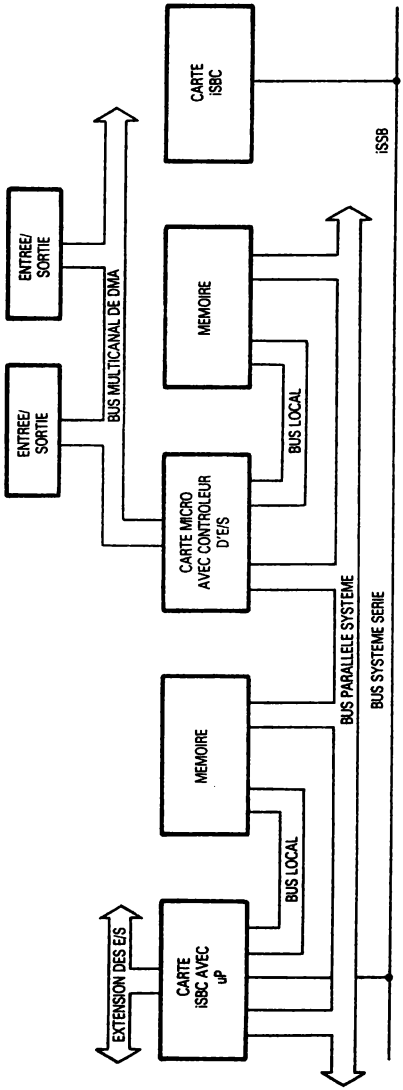


Fig. 5 — Le Multibus II de Intel se compose, en réalité, de 5 bus distincts.

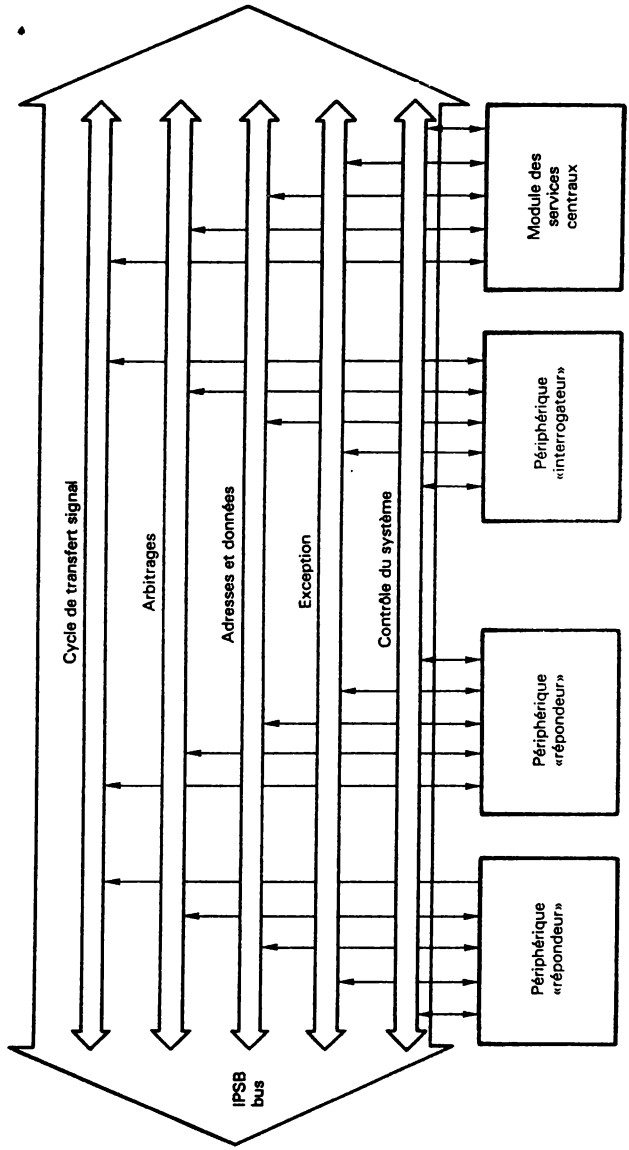


Fig. 6 — L'iPSB du Multibus II se subdivise, à son tour, en 5 bus.

Ces cinq bus apparaissent dans la figure 5. L'âme du système est l'iPSB, capable de travailler à 40 méga-octets par seconde et de desservir 20 cartes, appelées « agents » ; à son tour, il se subdivise en 5 bus (fig. 6) :

- adresses et données,
- arbitrage,
- contrôle système,
- cycle de transfert,
- cycle d'exceptions.

L'espace adressable par l'iPSB atteint 2^{32}, soit 64 tera-octets en mémoire, ce qui correspond à une adresse sur 32 bits ; il est de 2^{16} pour les E/S, soit 64 K, ainsi que 256 pour les messages et 32 pour l'interconnexion. Les adresses et les données sont multiplexées, ce qui explique de telles capacités pour un nombre total de fils réduits à 96 pour le seul iPSB. Le connecteur est un DIN, les cartes étant au format Eurocard dont nous donnons le schéma avec le bus VME (ce qui constitue un point commun pour ces bus).

Ce bus a été conçu pour les applications évoluées, multi-utilisateurs, telles que celles qui tournent sous Xenix ou iRMX (en temps réel), et pour la commande de périphériques rapides (contrôle de disques, de streamer, etc.). Via des lignes série RS-232, jusqu'à 16 utilisateurs peuvent avoir accès au système. La carte fond de panier peut recevoir jusqu'à 20 cartes.

La bande passante de l'iPSB est de 40 MHz, celle de l'iLBX, de 48 MHz. Les autres bus du système Multibus II offrent des bandes passantes inférieures ou égales à 10 MHz. L'ensemble Multibus II recourt au double connecteur DIN (IEC 603-2), partie de l'Eurocard standard.

CHAPITRE VII

BUS D'EXTENSION DES SYSTÈMES MICRO-ORDINATEURS

Une autre catégorie de bus est née des micro-ordinateurs et vise au développement des systèmes. Deux exemples types en sont fournis par les bus d'extension de l'Apple II et de l'IBM-PC.

Les mots-clés de ce chapitre

Bus d'extension de l'IBM-PC : caractéristiques, 134
Bus d'extension de l'Apple IIe : caractéristiques, 138
Bus LSI-11 : caractéristiques, 140
Bus d'extension de l'IBM-AT : caractéristiques, 142

1. LES BUS D'EXTENSION DE L'IBM-PC

BUS D'EXTENSION IBM-PC

Le bus d'extension de l'IBM-PC apparaît sur cinq connecteurs internes à la machine, câblés en parallèle. Il s'agit de connecteurs 62 broches, à broches espacées de 2,54 mm (100 « mils »). Les signaux qui y apparaissent sont, pour l'essentiel, ceux du microprocesseur de système, le 8088 ; y ont été adjoints des interruptions, accès direct à la mémoire, alimentations... en plus. La composition est la suivante (fig. 1) :

- 8 lignes bidirectionnelles pour les données,
- 20 lignes d'adresses,
- 6 niveaux d'interruptions,
- des lignes de commande pour la mémoire, de lecture/écriture, d'horloge,
- 3 canaux de DMA,
- des alimentations diverses,
- etc.

Adresses et données apparaissent sur des lignes distinctes : ce bus est démultiplexé. Quelques points caractéristiques sont à signaler. Ainsi, une ligne *Ready* sert aux opérations avec des circuits mémoires ou d'E/S insuffisamment rapides, selon un principe classique qui prolonge les temps d'accès dans un tel cas.

L'adressage des E/S, lui, se fait selon le concept de la distribution de l'espace adressable ; 512 E/S peuvent ici être adressées. Les affectations des broches du connecteur sont résumées dans le tableau ; le format des cartes qu'il peut recevoir est indiqué dans la figure 2.

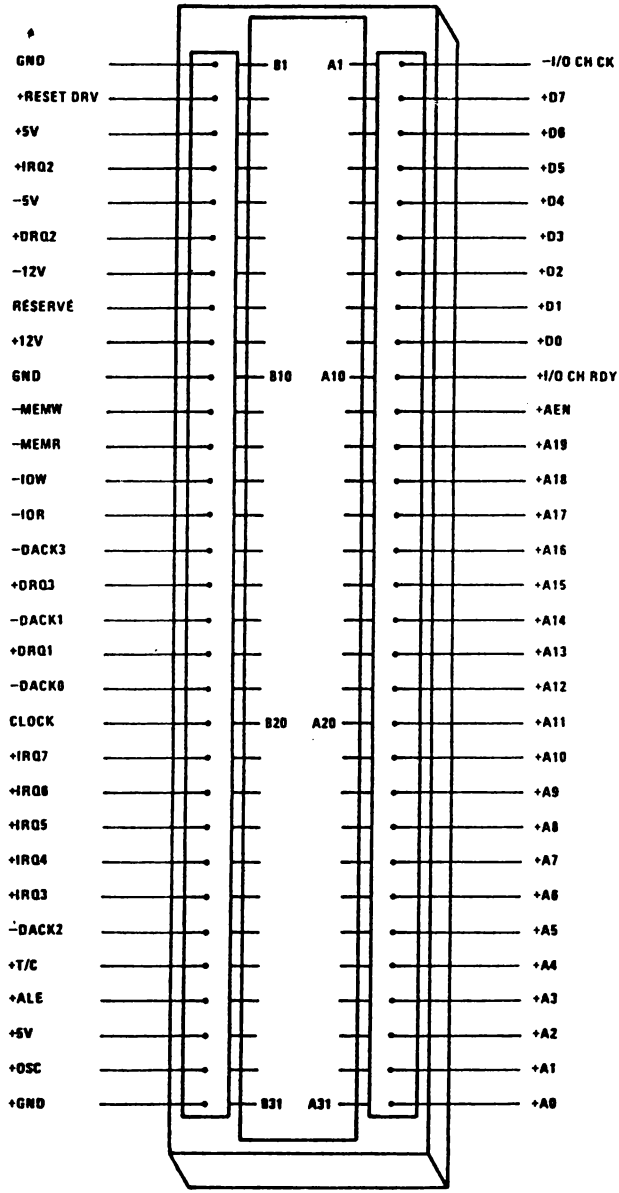


Fig. 1 — Le connecteur d'extension de l'IBM-PC, vu côté composants.

CONNECTEUR D'EXTENSION DE L'IBM-PC

Broche	Nom	Entrée ou Sortie	Fonction
A1	I/O CH CK	E	« I/O Channel Check » : test de parité
A2 à A9	D7 à D0	E/S	octet de donnée
A10	I/O CHRDY	E	« I/O Channel Ready » : pour E/S lentes
A11	AEN	S	« Adress Enable » : autorisation de DMA
A12 à A31	A 19 à A0	S	Adresse sur 20 bits
B1	GND	—	Masse
B2	Reset DRV	S	« Reset driver » : ré-initialisation du système
B3	+5V	—	Alimentation
B4	IRQ2	E	Demande d'interruption
B5	−5V	—	Alimentation
B6	DRQ2	E	Demande de DMA2
B7	−12V	—	Alimentation
B8	Réservé	—	Sur l'XT, sert à la sélection de carte
B9	12V	—	Alimentation
B10	GND	—	Masse
B11	MEMW	S	Ecriture mémoire
B12	MEMR	S	Lecture mémoire
B13	IOW	S	Ecriture E/S
B14	IOR	S	Lecture E/S
B15	DACK3	S	DMA 3 acquise
B16	DRQ3	E	Demande de DMA 3
B17	DACK1	S	DMA 1 acquise
B18	DRQ1	E	Demande de DMA 1
B19	DACK0	S	DMA 0 acquise
B20	CLK	S	Horloge système 4,77 MHz, rapport cyclique 33 %
B21 à B25	IRQ7 à IRQ3	E	Demandes d'interruptions 3 à 7
B26	DACK2	S	DMA 2 acquise
B27	T/C	S	Compte des terminaux pour DMA
B28	ALE	S	Ordre d'échantillonnage de l'adresse
B29	5V	—	Alimentation
B30	OSC	S	Oscillateur 14318,18 kHz, rapport cyclique 50 %
B31	GND	—	Masse

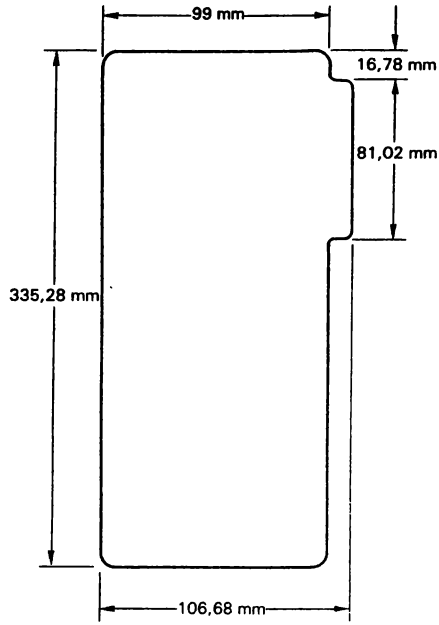


Fig. 2 — Format des cartes reçues par le connecteur d'extension de l'IBM-PC.

Pour mieux comprendre les sigles utilisés, notez que :

- **I vient de *in*, entrée (E)**
- **O vient de *out*, sortie (S)**
- **I/O vient de *in/out*, traduit par E/S**
- **R vient de *read*, lecture**
- **W vient de *write*, écriture**
- **IR vient de interruption.**

Voici quelques explications sur les attributions des lignes :

- A0 à A19 sont les 20 lignes d'adresses.
- D0 à D7 sont les mots de donnée, sur 8 bits.
- IRQ2 à IRQ7 sont les 6 lignes de demande d'interruption (niveaux 2 à 7).

- DRQ1 à DRQ3 sont les 3 lignes de demande d'accès direct en mémoire (DMA). DRQ1 a la plus haute priorité.
- DACK1 à DACK3 sont les 3 lignes d'acceptation de DMA.
- DACK0 est aussi une acceptation de DMA mais pour le rafraîchissement des mémoires dynamiques.
- OSC est un oscillateur rapide à 14318,18 kHz (période de 70 ns avec rapport cyclique de 30 %).
- CLK est l'horloge du système, à 4,77 MHz période de 210 ns et rapport cyclique de 30%.
- RESET DRV est le « reset » général du système.
- ALE valide l'adresse présente sur le bus.
- I/O CH CK signale une erreur de parité au niveau *bas*.
- I/O CH RDY prolonge le cycle de lecture avec des mémoires lentes ou des E/S lentes (au niveau *bas*).
- IOR et IOW sont les commandes de lecture et d'écriture pour les entrées-sorties.
- MEMR et MEMW sont les commandes de lecture et d'écriture en mémoire.
- T/C fournit une impulsion lorsque le compte des terminaux est atteint pour un canal de DMA.

Ajoutons que la broche B8, inutilisée sur le PC, sert à la sélection de cartes sur l'XT.

2. LE BUS D'EXTENSION DE L'APPLE IIe

BUS D'EXTENSION APPLE IIe

L'un des tout premiers micro-ordinateurs du marché, l'Apple II, né en 1975, a donné naissance à toute une famille de variantes (IIe, II+, IIc, etc.). Tous ces II, sauf le *c*, disposent de connecteurs d'extension (appelés « *slots* » en anglais, et souvent aussi en français !) destinés aux périphériques. En fait, ces connecteurs parallèles femelles de 50 broches chacun reprennent, pour l'essentiel, les accès du microprocesseur 6502 qui équipe les Apple II ; le 6502, lui, est un circuit intégré à 40 broches. La figure 3 donne le détail du brochage des connecteurs ; il s'agit de modèles référence Winchester #2HW25CØ-111, de 50 broches au pas de 0,1 pouce (2,54 mm). Les signaux essentiels sont définis dans le tableau.

CONNECTEUR D'EXTENSION POUR PERIPHERIQUES DE L'APPLE II

Broche	Nom	Fonction
1	I/O Select	Sélection d'une entrée-sortie
2 à 17	AØ à A15	Les 16 lignes du bus d'adresse
18	R/W	Lecture ou écriture
19	Sync	Synchronisation vidéo
20	I/O Strobe	Echantillonnage d'adresse entrée-sortie
21	RDY	« Ready » : pour mémoires lentes
22	DMA	Accès direct à la mémoire
23	Int Out	Sortie d'interruptions (connexion en chaîne)
24	DMA Out	Sortie d'accès directs en mémoire (chaînés)
25	+5V	Alimentation
26	Gnd	Masse
27	DMA In	Entrée d'accès directs en mémoire
28	INT In	Entrée d'interruptions
29	NMI	Interruption non masquable
30	IRQ	Demande d'interruption
31	RES	Reset, ré-initialisation
32	INH	Validation des ROM
33	−12V	Alimentation
34	−5V	Alimentation
35	Color Ref	3,5 MHz provenant du générateur vidéo
36	7M	Horloge 7 MHz
37	Q3	Horloge 2 MHz asymétrique
38	Φ	Phase 1 de l'horloge microprocesseur
39	User 1	Validation du décodage interne des entrées-sorties
40	ΦØ	Phase Ø de l'horloge microprocesseur
41	Device Select	Sélection de circuit
42 à 49	DØ à D7	Les 8 fils du bus de données
50	+12V	Alimentation

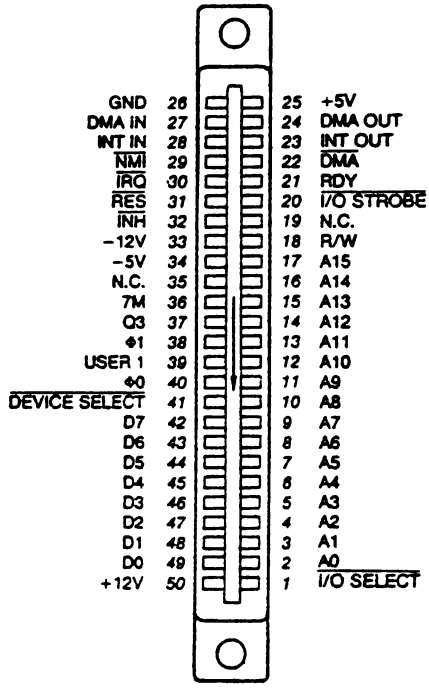


Fig. 3 — Le connecteur d'extension pour périphérique des célèbres Apple II.

3. LE BUS DU LSI-11

BUS LSI-11

Digital Equipment Corporation (DEC) a annoncé le micro-ordinateur LSI-11 en 1975, créant un bus spécialisé 16 bits bidirectionnel, avec multiplexage données/adresses. Il supporte en version minimale 6 charges, intermédiaire 12 charges, et maximale 15 charges sur une longueur de 25 cm. La carte même comporte deux groupes de broches, repérés par les lettres A, et B (en module double, elles sont désignées par A, B, C et D). Chaque groupe représente 36 broches, dont les affectations sont résumées dans le tableau ; les broches notées 1 se trouvent *côté composants,* celles marquées 2 *côté circuit.*

BUS DU LSI-11 DEC

Groupe	Broche	Nom	Fonction
A1	AA1 et AB1	BSPARE 1 (et 2)	Réserve
	AC1 et AD1	BAD 16 (et 17)	Bit d'extension adresse
	AE1 à AH1	SPARE 1 (à 3)	Réserve spéciale
	AJ1	GND	Masse signal
	AK1 et AL1	MSPAREA	Réserve maintenance
	AM1	GND	Masse signal
	AN1	BDMRL	Demande de DMA
	AP1	BHALTL	Halte processeur
	AR1	BREFL	Rafraîchissement mémoire
	AS1	+12B	Alimentation 12 V
	AT1	GND	Masse signal
	AU1	PSPARE1	Réserve
	AV1	+5B	Alimentation 5 V
A2	AA2	+5	Alimentation 5 V
	AB2	−12	Alimentation −12 V
	AC2	GND	Masse signal
	AD2	+12	Alimentation 12 V
	AE2	BDOUTL	Sortie donnée
	AF2	BRPLYL	Réponse
	AH2	BDINL	Entrée donnée
	AJ2	BSYNCL	Synchronisation
	AK2	BWTBTL	Ecriture octet
	AL2	BIRQL	Demande d'interruption
	AM2	BIAKIL	Entrée acquisition interruption
	AN2	BIOKOL	Sortie acquisition interruption
	AP2	BBS7L	Sélection banque 7
	AR2	BDMGIL	Entrée de DMA accordée
	AS2	BDMGOL	Sortie de DMA accordée
	AT2	BINITL	Initialisation
	AU2	BDALØL	Ligne adresse/donnée
	AV2	BDAL12	Ligne adresse/donnée
B1	BA1	BDCOKH	Alimentation continue OK
	BB1	BPOKH	Alimentation OK
	BC1 à BH1	SSPARE 4 (à 8)	Réserve spéciale
	BJ1	GND	Masse signal
	BK1 et BL1	MSPAREB	Réserve maintenance
	BM1	GND	Masse signal
	BN1	BSACKL	Réponse DMA élément
	BP1	BSPARE6	Réserve
	BR1	BEVNTL	Interruption externe
	BS1	PSPARE 4	Réserve
	BT1	GND	Masse signal
	BU1	PSPARE2	Réserve
	BV1	+5	Alimentation 5 V
B2	BA2	+5	Alimentation 5 V
	BB2	−12	Alimentation −12 V
	BC2	GND	Masse signal
	BD2	+12	Alimentation 12 V
	BE2 à BV2	BDAL2L à BDAL15L	14 lignes donnée/adresse

4. LE BUS D'EXTENSION DU PC-AT D'IBM

BUS D'EXTENSION DE L'AT D'IBM

Conçu autour du microprocesseur 80286, l'AT d'*IBM* est à la fois plus puissant et plus rapide que les PC et XT. Il dispose de 14 connecteurs d'extension (fig. 4) répartis en deux groupes de 62 et 36 broches. Ils supportent :

- Un bus d'adresse sur 24 bits, adressant donc jusqu'à 16 méga en mémoire.
- Un bus de données sur 8 ou 16 bits.
- Des interruptions, DMA, rafraîchissement de mémoire, etc.

La figure 5 montre l'organisation du connecteur 62 broches et la figure 6, celle du connecteur 36 broches, ainsi que les noms des signaux dont la signification est présentée dans le tableau.

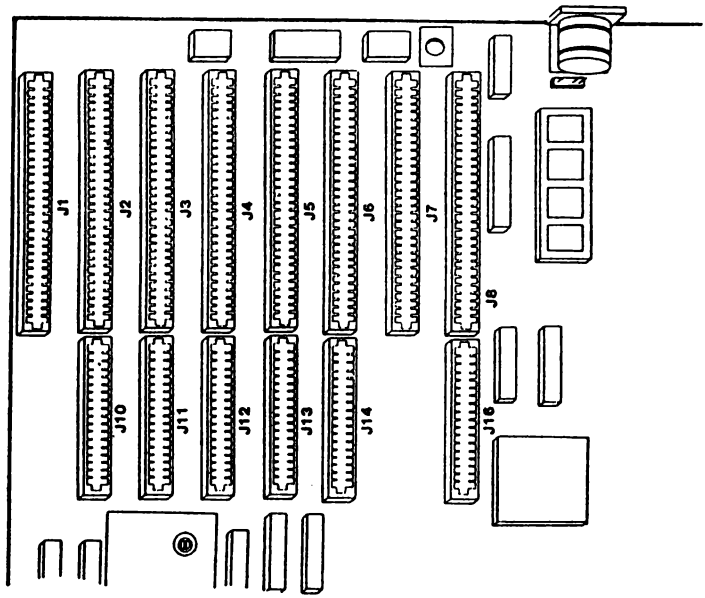


Fig. 4 — Les connecteurs d'extension de l'AT d'IBM.

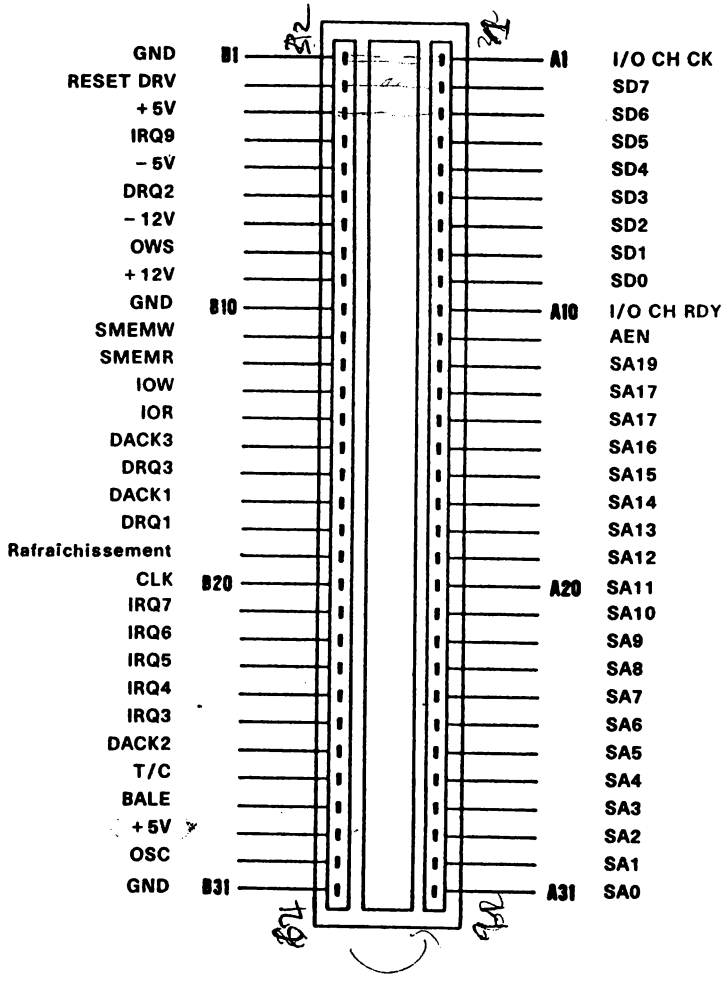


Fig. 5 — Le connecteur d'extension 62 broches de l'AT et ses signaux ; il est vu côté composants.

CONNECTEURS D'EXTENSION DE L'AT

Signaux	Fonctions
SAØ à SA19	20 lignes d'adresses pour mémoire et E/S (entrée/sortie)
LA17 à LA23	Lignes d'adresses complémentaires
BALE	Echantillonnage de l'adresse, mémorisation, validation
CLK	Horloge système à 6 MHz, cycle de 167 ns à 50 %
RESET DRV	Reset de la logique à la mise en service du système
SDØ à SD15	Données
I/O CH CK	Erreur de parité
I/O CH RDY	Prolongation des cycles mémoire ou E/S par pas de 167 ns et jusqu'à 2,5 µs
IRQ3 à IRQ15	Demandes d'interruption
IOR	Lecture d'E/S
IOW	Ecriture d'E/S
SMEMR	Lecture mémoire pour moins de 1 Mo
SMEMW	Ecriture mémoire pour moins de 1 Mo
MEMR	Lecture mémoire pour champ complet
MEMW	Ecriture mémoire pour champ complet
DRQØ à DRQ7	Demandes de DMA
DACKØ à DACK7	Acquisition de demande de DMA
AEN	Adressage en DMA autorisé
Rafraîchissement	Pour mémoires dynamiques
T/C	Compte des terminaux pour DMA
SBHE	Octet fort des données (sur 16 bits) autorisé
MASTER	Pour mode interruption en DMA
MEM CS 16	Transfert en E/S sur 16 bits
OSC	Oscillateur 14318,18 kHz, période de 70 ns à rapport cyclique de 50 %
ΦWS	Pas d'états d'attente

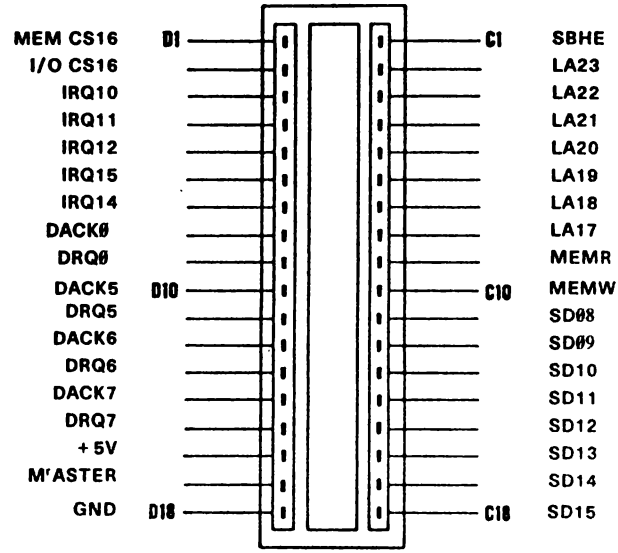


Fig. 6 — Le connecteur d'extension 36 broches de l'AT, vu côté composants, et ses signaux.

CHAPITRE VIII

LIAISONS AVEC LES DISQUES

D'autres normes, souvent de fait, ont pu être établies afin de relier les unités centrales aux unités à disque fixe (dur) ou à disquettes. En voici un aperçu.

Les mots-clés de ce chapitre

SA450 : norme pour disquettes, 148
ST506 : norme pour disques, 148 et 161
ESDI : norme pour disques durs, 148 et 161
SASI : norme pour disques, 148

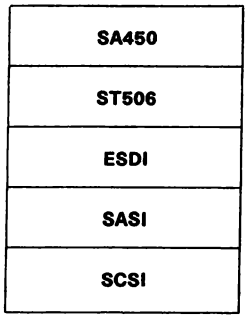
SCSI : norme pour disques, 148
Unité à disquettes 5"1/4 : caractéristiques, 151
Unité à disquettes 3"1/2 : caractéristiques, 152
Microdisquette 3"1/2 : caractéristiques, 150

Bus pour disquettes, 153
Circuits intégrés pour disques, 154
Circuit WD 279X : CI pour disques, 154
Disquettes IBM-PC, 156

Formats pour disquettes IBM, 160
XT d'IBM : liaison au disque fixe, 165
Macintosh et disques, 166

1. INTERFACES POUR DISQUETTES ET DISQUES

De nombreuses interfaces ont été étudiées afin de relier des unités à disquettes ou disques durs à l'ordinateur. La figure 1 en situe quelques-unes, aussi bien pour micros que minis ou que pour disquettes et disques durs, d'ailleurs. Dans cette figure :



• La SA450 a pour origine *Shugart* ; c'est probablement la norme la plus populaire pour les disquettes 5"1/4, double face. Elle a également été adoptée pour les 3"1/2.

• La ST506 est devenue l'interface de fait pour les disques Winchester 5"1/4, durs par conséquent. A son origine se trouve *Seagate Technology*. La ST506 existe aussi avec des Winchesters 3"1/2.

• L'ESDI *(« Enhanced Small Device Interface »)* constitue une amélioration de la ST506, pour disques durs 5"1/4 de hautes performances. Elle accepte un débit supérieur à 5 Mbits/s (cas de la ST506). Les câbles sont très proches.

• La SMD (Storage Module Drive), introduite par *CDC*, couvre les unités à disques de 5"1/4 à 14 pouces.

• La SASI est due à *Shugart* (SASI = *« Shugart Associates System Interface »*). Elle s'applique aux disquettes et disques durs.

• La SCSI constitue un développement de la SASI. Le sigle provient de *« Small Computer System Interface »*. Elle a été normalisée sous le code ANSI de X3T9.2.

En pratique, on retrouvera les SA450 et ST506/412 pour les 3"1/2, l'ESDI en plus pour les 5"1/4.

INTERFACES POUR DISQUES

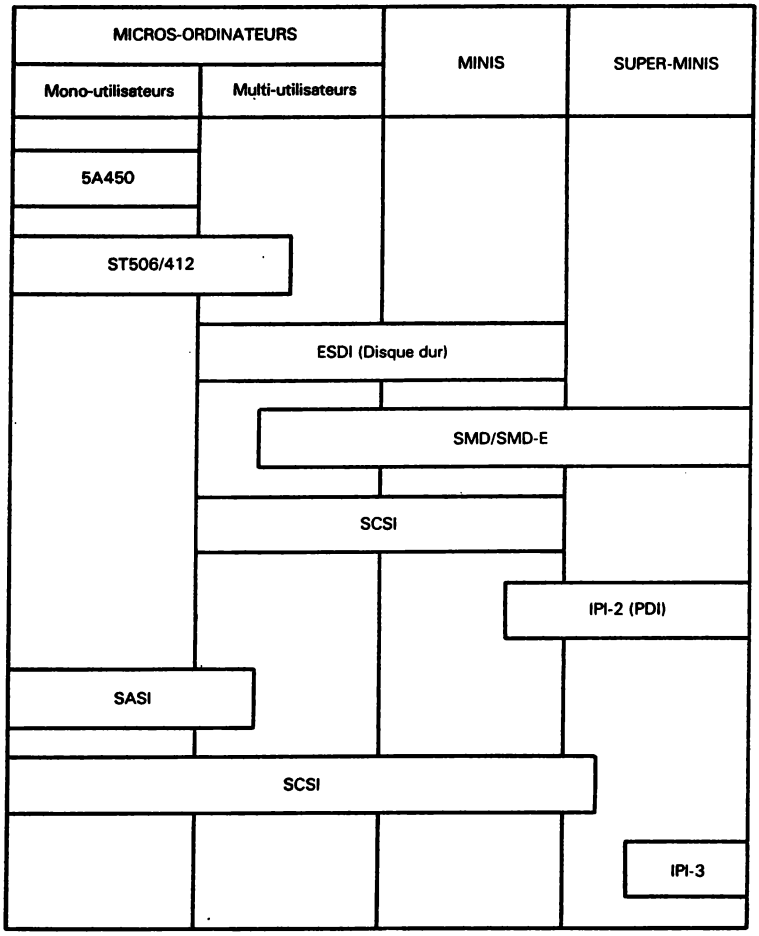


Fig. 1 — Quelques-unes des grandes interfaces pour disquettes et disques, avec leur zone d'application en micro et mini-informatique.

2. BUS POUR UNITÉS A DISQUETTES

Une unité à disquettes constitue un ensemble relativement complexe, dont les principales caractéristiques sont :

- Diamètre des disquettes : 5''1/4 (environ 13 cm, un pouce valant 2,54 cm), 3''1/2 (moins de 9 cm), et « anciennement » 8'' (20 cm environ).
- Simple face, ou double face, et par conséquent avec simple ou double jeu de têtes de lecture-écriture.
- Densité d'enregistrement, nombre de pistes, de secteurs, etc.
- Format d'enregistrement, capacité, etc.

La figure 3 propose une structure type d'unité à disquettes et la figure 4 l'équivalent pour des microdisquettes 3''1/2, selon *Shugart*. La figure 2 rappelle l'aspect d'une telle microdisquette.

MICRODISQUETTES
3''1/2

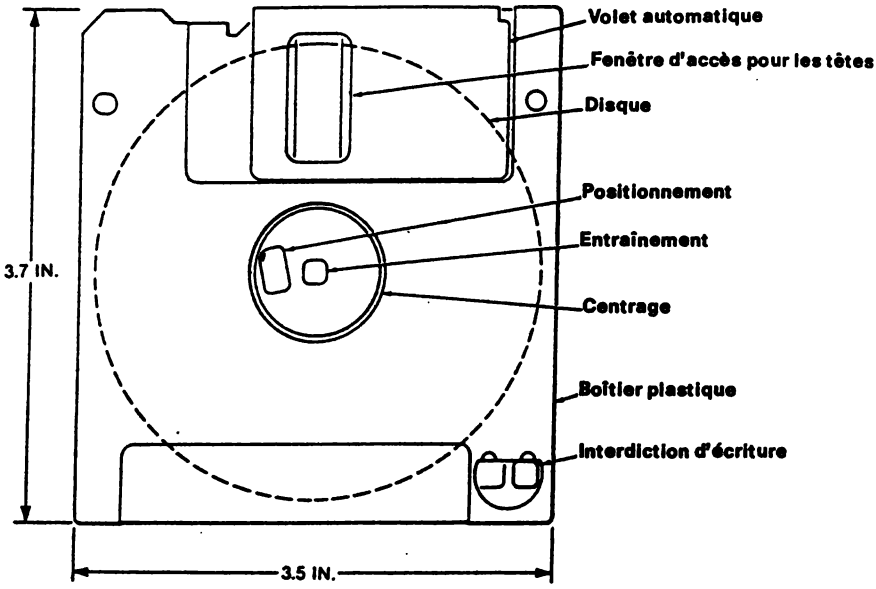


Fig. 2 — Aspect d'une microdisquette 3''5.

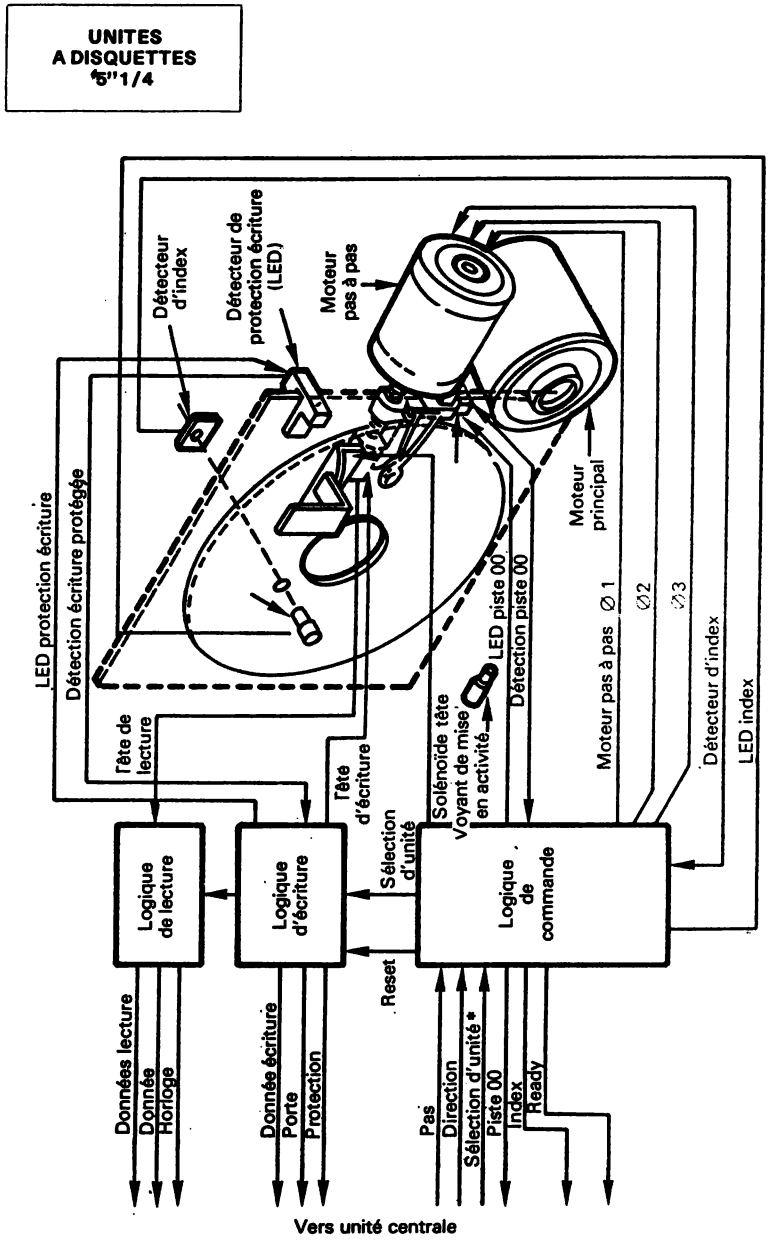


Fig. 3 — L'électronique de l'unité à disquettes 5"1/4.

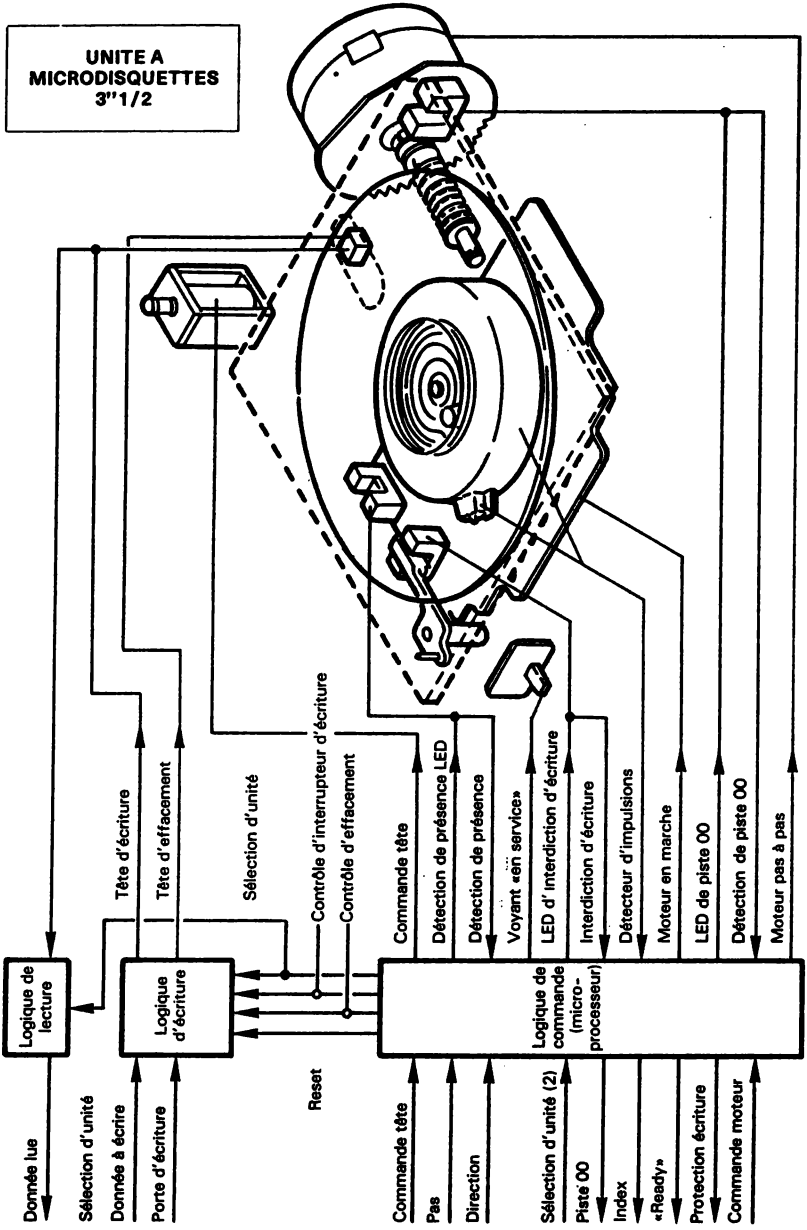
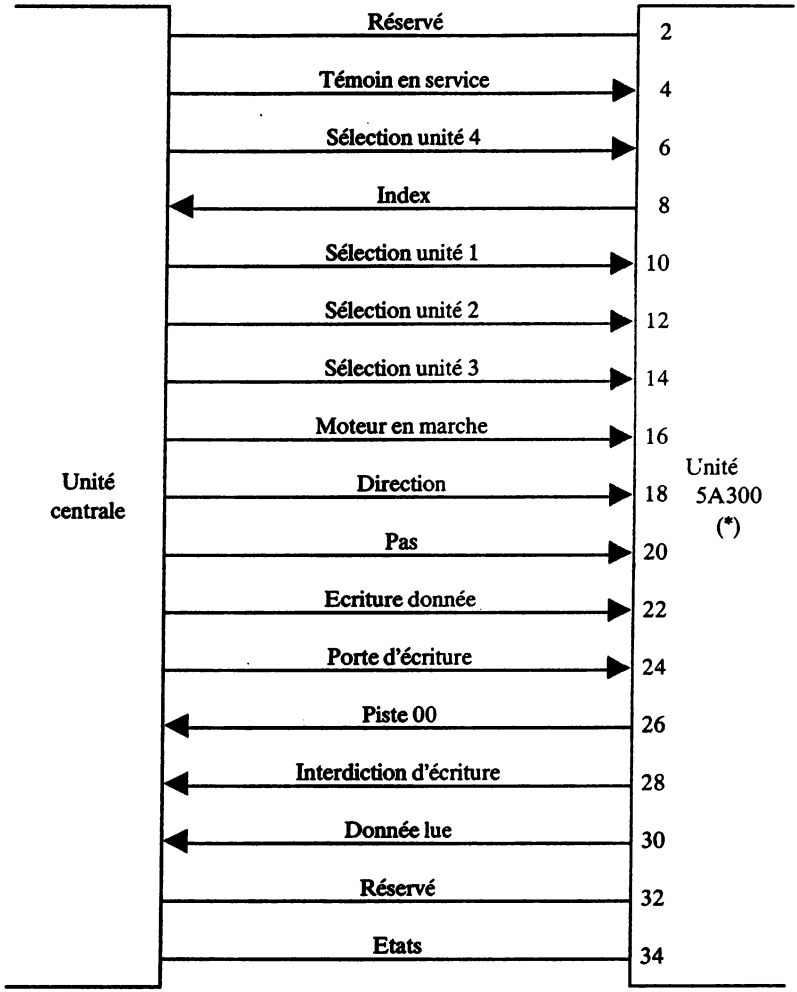


Fig. 4 — Unité à microdisquettes SA300 de Shugart.

BUS POUR DISQUETTES

L'interface avec l'unité centrale fait généralement appel à un connecteur 34 broches, identique très souvent en 5"1/4 et 3"1/2, ce qui rend les interfaces « universelles ». La figure 5 montre ce connecteur, vu du côté de l'unité à disquettes (ici, la SA300) ; tous les contacts d'ordre *impair* sont connectés à la masse.



(*) Tous les contacts impairs (de 1 à 33) sont reliés à la masse.

Fig. 5 — Le connecteur 34 broches pour interfacer une unité à disquettes ou microdisquettes.

Un second connecteur servira à l'alimentation + 12 V et + 15 V, dans le cas de la SA300. La ligne de « direction » commande le sens du mouvement des têtes (vers le centre ou le bord), le déplacement étant commandé par pas, via la ligne « pas » transmettant des impulsions de 1 µs minimum, séparées par 6 µs au moins. La ligne 26 fournit l'indication du positionnement des têtes sur la piste 00, la plus extérieure, afin que la logique ne risque pas d'ordonner un nouveau déplacement vers l'extérieur. La ligne 34 indiquera, au choix et en fonction de la position d'un cavalier, si une disquette a été insérée dans l'unité ou si, en outre, le moteur tourne et les têtes se trouvent sur la piste spécifiée.

La mise en service du moteur se commande par la ligne 16 ou par l'une des lignes de sélection d'unités. Le codage est assuré en FM ou MFM, techniques présentées dans un chapitre spécial consacré aux codes binaires.

3. CIRCUITS INTÉGRÉS D'INTERFACE

CIRCUITS INTEGRES POUR DISQUES

CIRCUIT WD 279X

CIRCUITS INTEGRES D'INTERFACE DISQUETTES

De nombreux fabricants de circuits intégrés proposent des circuits spécialisés exécutant toutes les fonctions d'interfaçage, y compris la modulation MF et MFM. Ainsi, *NEC, Western Digital*, etc. Prenons l'exemple des WD 279X de *Western Digital*. Ces circuits, compatibles *IBM* ou *Shugart*, s'appliquent ainsi à la commande de très nombreuses unités à disquettes 5"1/4 ; ils comprennent un séparateur de données, une horloge pour codage en simple densité (modulation en fréquence) ou double densité (modulation de fréquence modifiée : MFM), un générateur et testeur de redondance cyclique (test d'erreur), un oscillateur, un diviseur et un formateur logiciel, plus des registres comptant les pistes et les secteurs, etc.

L'interfaçage d'un processeur et d'une unité à disquettes ne requiert ainsi, outre le 279X, que quelques composants passifs (fig. 6). Elle se fait à partir du circuit de commande d'*accès direct à la mémoire* (DMA, ou ADM) puisque les liaisons procèdent selon ce mode d'échange.

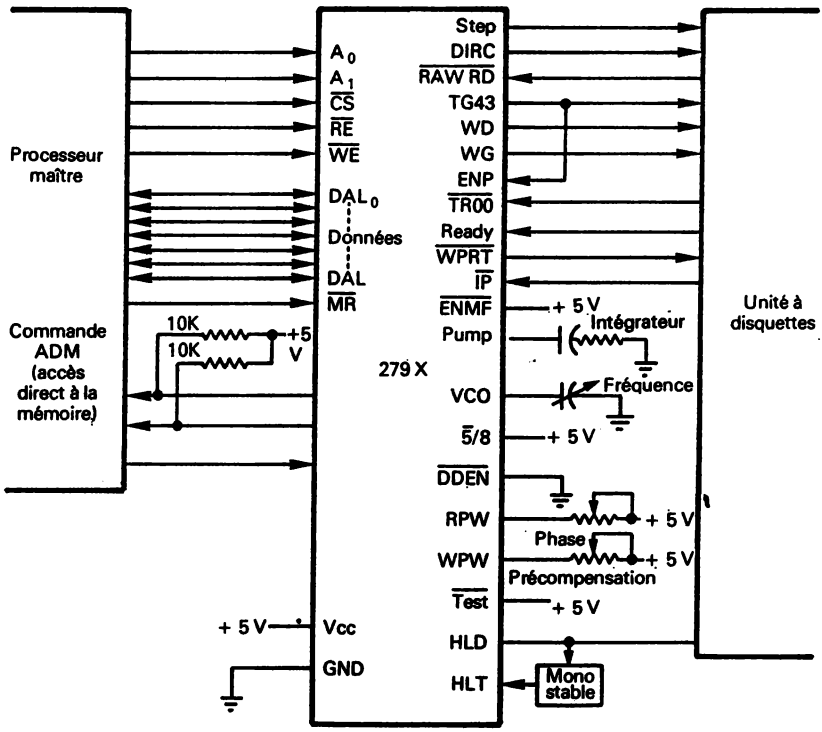


Fig. 6 — Intervention de l'interface WD 279X pour commande d'une unité à disquettes.

4. UNITÉS A DISQUETTES DE L'IBM-PC

DISQUETTES DE L'IBM-PC

L'IBM-PC offre une ou deux unités à disquettes 5"1/4 double face, de 320 K octets chacune (DOS 1.), à 8 secteurs par piste, ou 360 K octets (DOS 2.) avec 9 secteurs par piste, pour un total de 40 pistes. Chaque secteur représente 512 octets. Un calcul rapide montre ainsi la capacité d'une disquette à 9 secteurs :

9 secteurs × 40 pistes × 512 octets par secteur × 2 faces = 368 640 octets

soit 360 K. La densité des pistes est de 48 pistes par pouce (elle passe à 96 pour l'AT). La sectorisation est logicielle.

La modulation est du type MFM (modulation de fréquence modifiée) dont nous présentons les caractéristiques dans un chapitre spécial consacré aux techniques de modulation.

Introduite dans l'un des connecteurs d'extension internes, la carte d'interface pilote deux unités à disquettes via un câble plat, en chaîne. Un second connecteur permet l'association d'unités externes. L'interface est conçue autour du circuit intégré spécialisé µPD765 de *NEC* ; le transfert des données s'effectue via le canal de DMA (accès direct en mémoire) du système ; un niveau d'interruption témoigne de l'exécution des transferts. Le principe de la carte d'interface est donné figure 7. Le µPD765 est, bien sûr, programmable.

Un connecteur d'extrémité de carte d'interface, de 34 contacts, reçoit le câble de liaison. Les contacts d'ordre *impair* sont situés côté circuit, *pair* côté composants, avec les attributions notées dans la figure 8.

La disquette est introduite dans son unité alors que le volet de protection est relevé. Le rabattre provoque l'insertion du cône d'entraînement central, tournant à vitesse constante de 300 tours par minute sous commande d'un servomoteur à courant continu. En même temps, les têtes entrent en contact avec la disquette ; elles sont ensuite positionnées grâce à un moteur pas à pas quadriphase ; un pas correspond à un mouvement linéaire d'une piste.

L'écriture crée une piste de 0,33 mm de largeur, ramenée par effacement à 0,3 mm (0,012 pouce). L'unité dispose également de :

- un système de détection de la position des têtes sur la piste Ø ;
- un index à LED et phototransistor détectant le trou d'index ;
- une détection d'interdiction d'écriture.

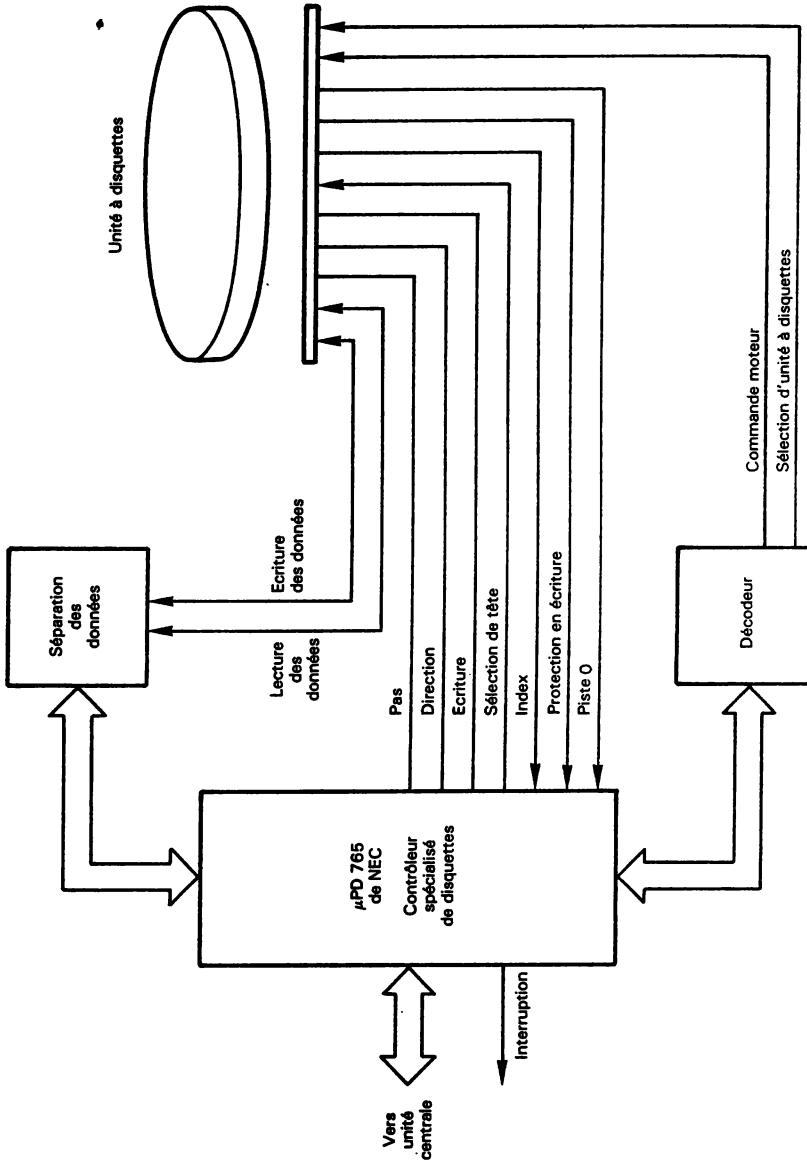


Fig. 7 — Principe de la commande de l'unité à disquettes avec un µPD765 de NEC, sur l'IBM-PC, pour une seule unité.

INTERFACE DISQUETTES IBM-PC

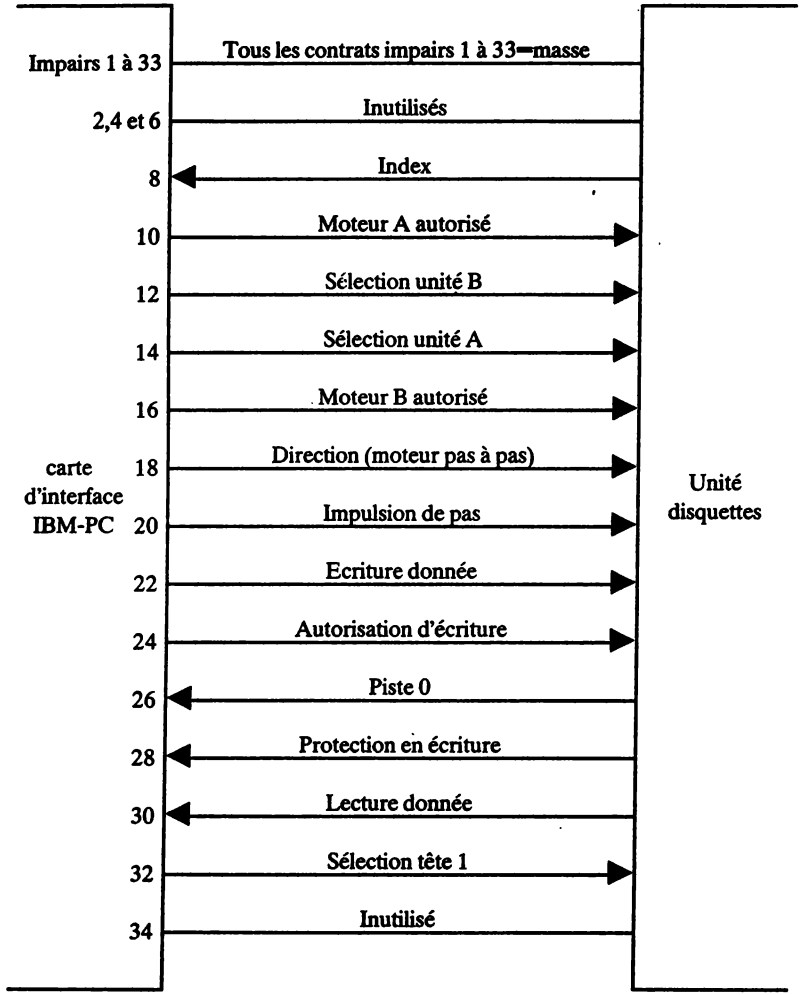


Fig. 8 — La liaison interface à unités à disquettes, sur l'IBM-PC. Tous les niveaux sont TTL. Les contacts d'ordre pair sont du côté composant de la carte d'interface.

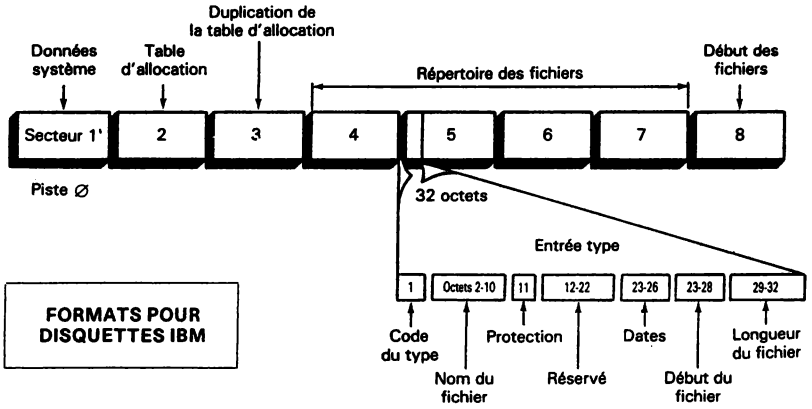
D'un diamètre exact de 133,4 mm (5''1/4), la disquette tourne librement et est lubrifiée dans sa jaquette de protection. Les caractéristiques générales sont résumées dans le tableau.

CARACTERISTIQUES DES DISQUETTES 5''1/4 DE L'IBM-PC

Caractéristiques	Valeurs
Diamètre	5''1/4, soit 133,4 mm
Pistes par pouce Nombre de pistes	48 40
Faces Nombre de secteurs	2 (double face) 8 ou 9, selon DOS
Capacité	320 ou 360 Ko, selon DOS
Température • de fonctionnement • de stockage	 10 à 44°C −40 à 60°C
Humidité relative • en fonctionnement • en stockage	 20 à 80 % 5 à 95 %
Temps de recherche de piste à piste Temps de positionnement des têtes Taux d'erreur	8 ms 25 ms 1 par 10^9 (récupérable) 1 par 10^{12} (non récupérable) 1 par 10^6 (recherche)
Durée de vie des têtes Durée de vie du support	20 000 heures (usage normal) 3.10^6 passages par piste
Vitesse de rotation Variations instantanées de la vitesse Temps de start/stop Vitesse de transfert	300 tours/minute ± 1,5 % ± 3 % 500 ms max 250 Kbits/s
Mode d'enregistrement	MFM
Consommation d'une unité	900 mA sous 12 V plus 600 mA sous 5 V en moyenne

Le *format d'enregistrement* est donné dans la figure 9, et comparé d'ailleurs au célèbre « IBM 3741 » pour les disquettes 8". L'interface disquette est généralement organisée autour d'un circuit spécialisé *NEC* ou *Intel*, type 765, ou *Western Digital* type 179X, avec contrôleur 2903. Le code de correction d'erreur est le CRC, aux normes CCITT (que nous analysons dans un chapitre spécifique).

A — Disquette 5"1/4 IBM-PC
40 pistes, version 8 secteurs de 512 octets



B — Format IBM 3741, disquettes 8"
77 pistes, 26 secteurs de 128 octets

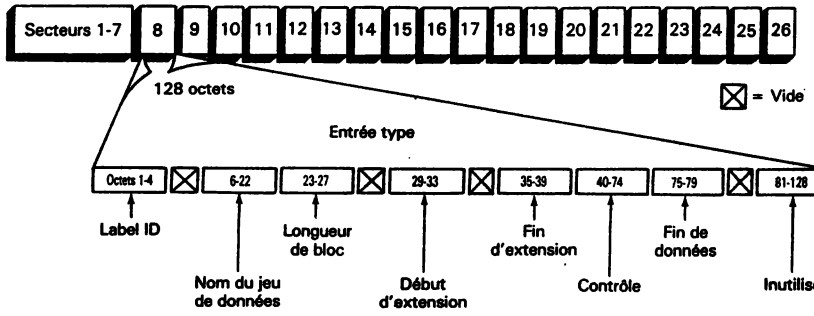


Fig. 9 — Format des données sur la piste 00 de disquettes 5"1/4 (en A, pour l'IBM-PC) ou 8" (format 3741 IBM).

5. INTERFACES POUR DISQUE DUR ST506 ET ESDI

ST 506

ESDI

Pour gérer les disques durs 5 pouces 1/4, la société *Seagate* a créé une interface, normalisée par la suite sous la référence ST 506/412. Deux câbles sont utilisés, avec :

- un connecteur 34 broches, pour les signaux de commande, d'ailleurs identiques à ceux des disquettes ;
- un connecteur 20 broches pour les données.

La séparation des données a été rendue indispensable car la vitesse de transmission atteint 5 mégabits par seconde. Pour la même raison, les liaisons sont symétriques.

Le connecteur 34 broches est semblable à celui utilisé pour les disquettes (fig. 10) ; toutes les broches d'ordre *impair*, côté circuit, sont reliées à la masse. La longueur maximale de câble est de 20 pieds (environ 7 m).

Le même câble peut piloter jusqu'à 4 unités à disques fixes, reliées en chaîne. Par contre, la liaison des données sera personnalisée, en étoile.

La figure 11 donne le détail de la liaison 20 broches ; toutes les broches d'ordre *pair, sauf les 14 et 18*, sont reliées à la masse.

NORME ESDI POUR DISQUE DUR

L'interface ESDI est très proche du ST 506/412 et recourt aux mêmes connecteurs 34 et 20 broches. Elle le développe toutefois et permet l'utilisation du codage NRZ1 (étudié dans un chapitre spécial). Avec l'ESDI, c'est aussi l'unité à disque et non le contrôleur qui fournit l'horloge. En outre, le positionnement des têtes peut être commandé en mode série et non plus pas à pas, ce qui permet un déplacement plus rapide sur le cylindre visé. Les liaisons ESDI sur 34 broches et 20 broches sont représentées, dans leur principe, dans la figure 12.

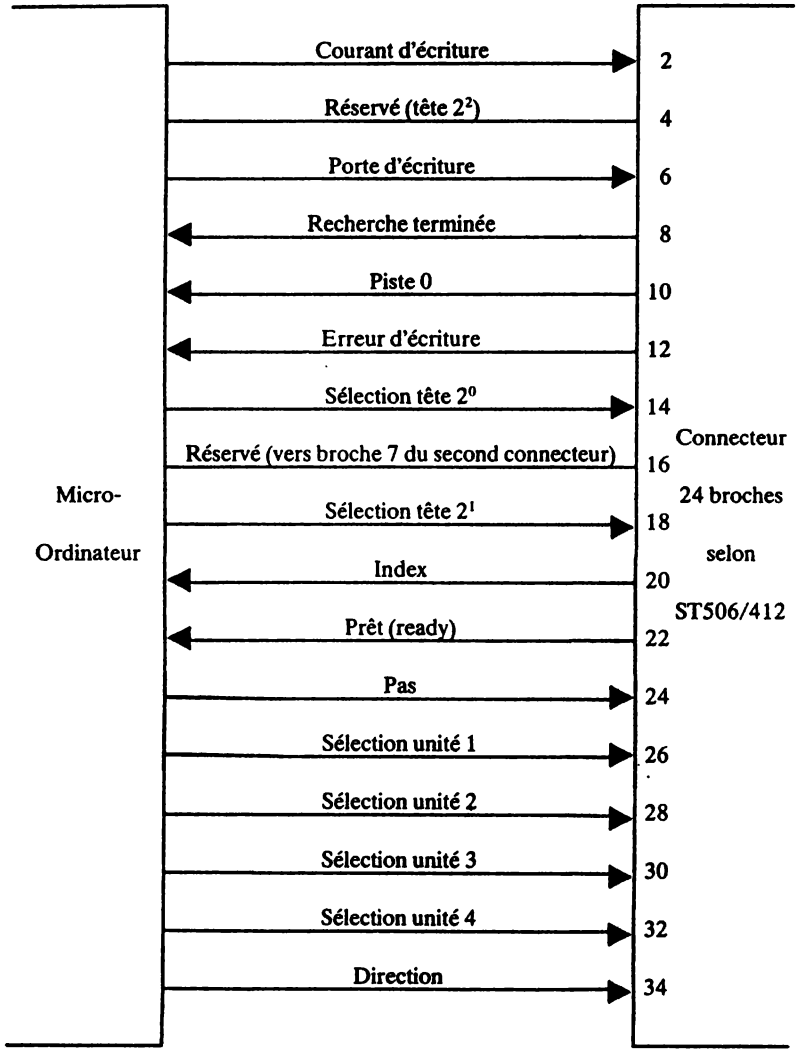


Fig. 10 — La liaison à un disque dur 5"1/4 selon la norme ST 506/412 : le connecteur 34 broches. Toutes les broches d'ordre impair sont connectées à la masse.

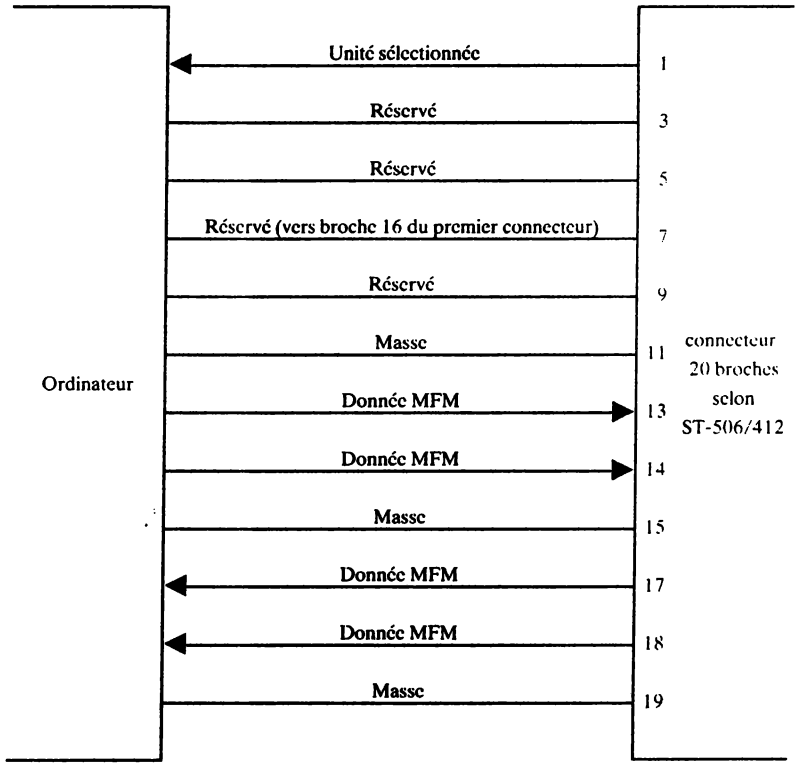
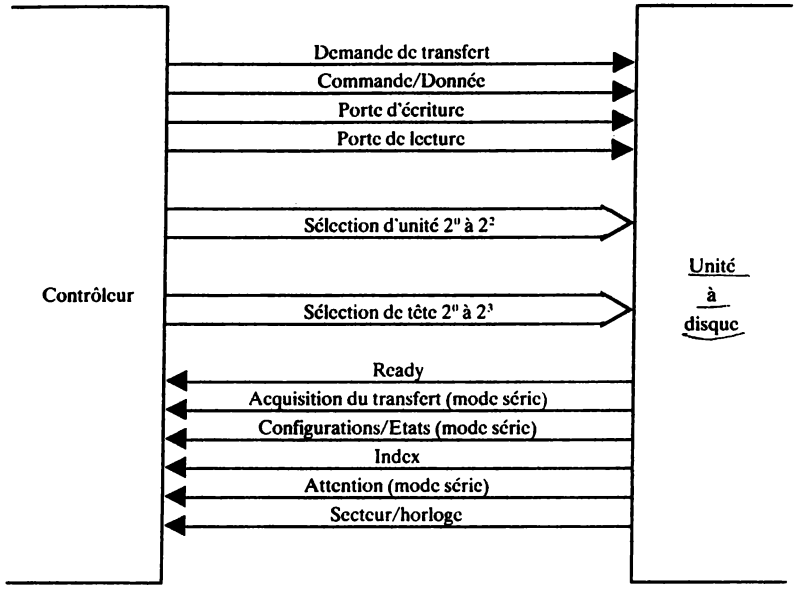


Fig. 11 — La liaison à un disque dur 5"1/4 pour les données, sur le second connecteur 20 broches en ST 506/412. Toutes les broches d'ordre pair, *sauf* 14 et 18, sont reliées à la masse.

CABLE 34 BROCHES



CABLE 20 BROCHES

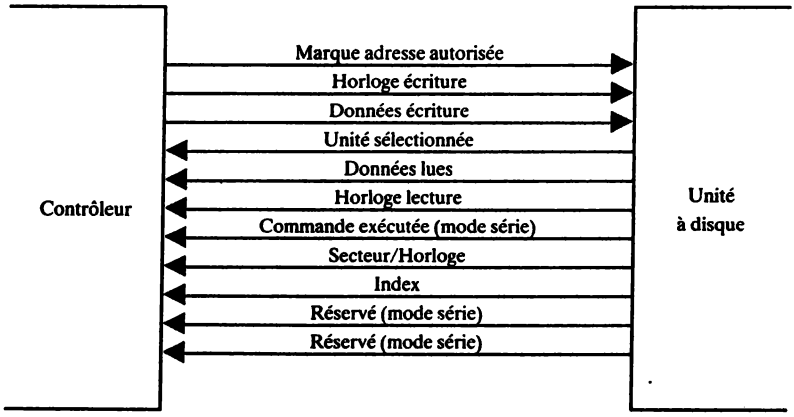


Fig. 12 — Principe de la liaison au disque dur selon la norme ESDI, pour une commande « série » des actionneurs des têtes.

6. LIAISON DU DISQUE 10 M DE L'IBM PC-XT

XT D'IBM : LIAISON AU DISQUE FIXE

Le micro-ordinateur PC-XT d'*IBM* dispose d'une carte d'interface capable de gérer deux unités à disque fixe 10 M, bien que le boîtier unité centrale n'en comprenne qu'une. Elle offre un connecteur 34 broches (J1) pour les signaux de commande, cascadables, et deux connecteurs 20 broches pour les données de deux unités (J2 et J3). On retrouve, là, l'organisation définie par la norme ST 506.

La broche 5 des connecteurs 20 broches est absente et sert au détrompage.

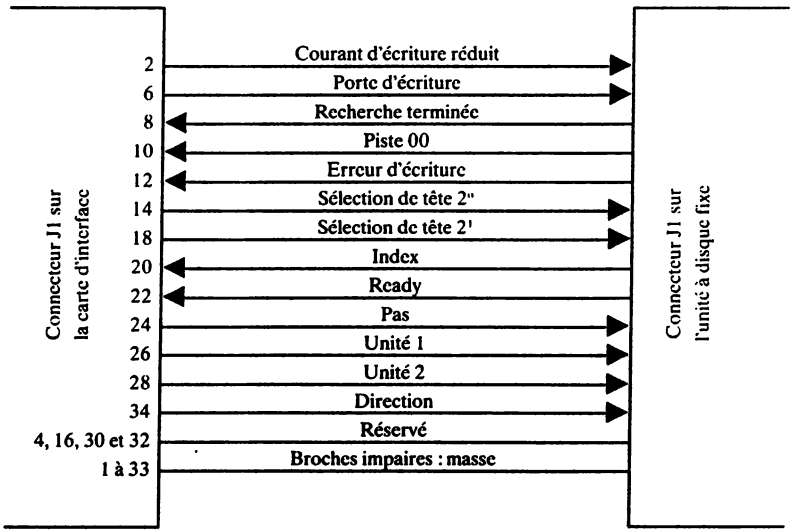


Fig. 13 – La liaison 34 broches au disque fixe avec l'IBM-XT.

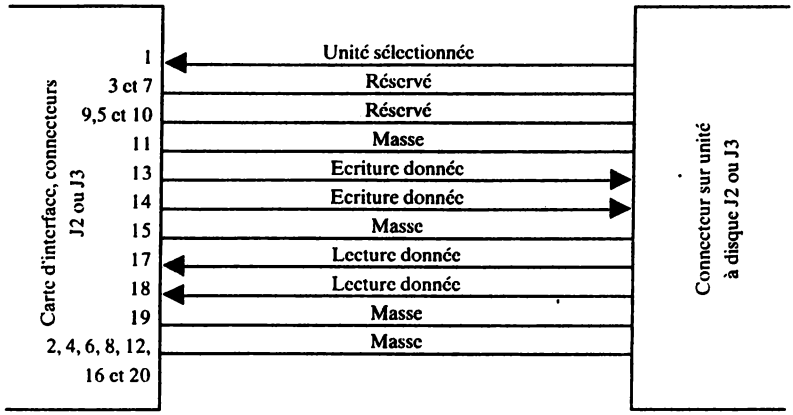


Fig. 14 — Liaison 20 broches au disque fixe sur l'IBM-XT.

La liaison est assurée ainsi que le montrent les figures 13 et 14. La détection et la correction des erreurs sont assurées en mode ECC sur 32 bits. L'unité à disque fixe utilise *deux* plateaux de 5''1/4, avec une tête mobile par surface pour 306 cylindres. Chaque piste est divisée en 17 secteurs de 512 octets, avec un total de 1 224 pistes. La capacité formatée est de 10 M. Un air filtré (filtre à 0,3 µm) circule dans l'unité. Les disques tournent à 3 600 tours/minute ; le temps de latence moyen est de 8,33 ms pour un temps d'accès piste à piste de 3 ms. La vitesse de transfert est de 5 Mbits/s, la modulation est MFM et la durée de vie est prévue pour au moins 5 ans (8 000 heures de MTBF).

7. LA LIAISON DISQUE DU MACINTOSH

MACINTOSH ET DISQUES

Le Macintosh d'*Apple* peut être connecté à une unité à disque externe à l'aide d'une liaison partant d'un connecteur DB-19 sur la machine. La figure 15 illustre la liaison à une unité *Sony*, d'ailleurs effectuée via un câble 20 fils.

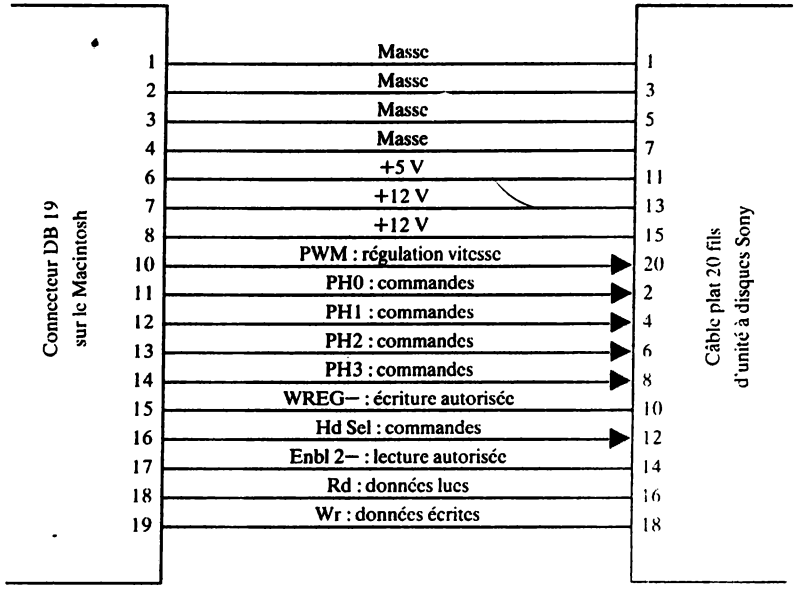


Fig. 15 — Liaison Macintosh à unité à disques Sony.

Dans cette liaison ne sont pas utilisées les broches 5 et 9 du connecteur DB-19 ; la broche 5 correspondant au – 12 V.

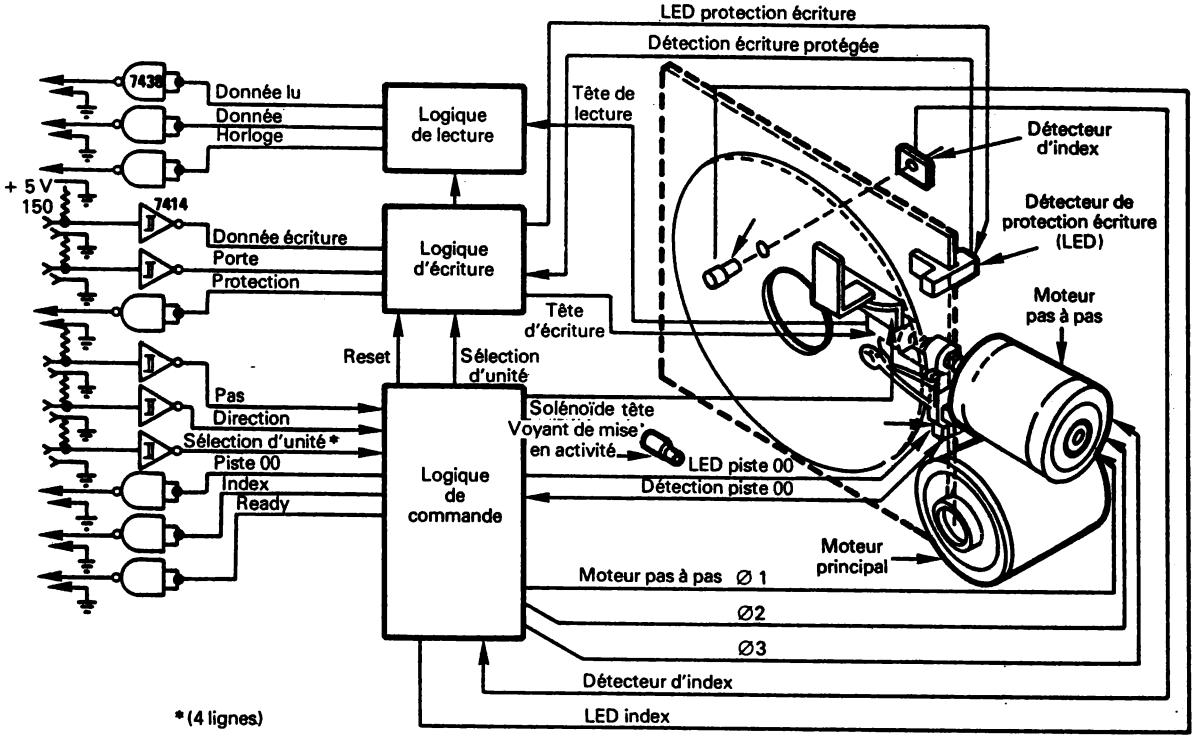


Fig. 6. — L'électronique de l'unité à disquettes.

La liaison devrait être asynchrone, à des vitesses atteignant 1,5 méga-octets par seconde et sur 15 mètres. L'adressage se ferait sur 32 bits, le câble SASI se terminant par un connecteur à 50 broches.

4.1. Interface pour unités à disquettes

Pour interfacer des unités à disquettes, une société s'est particulièrement distinguée, *Western Digital,* en réalisant des circuits intégrés complexes spécifiques. Les WD 279X, compatibles *IBM* ou *Shugart,* s'appliquent ainsi à la commande de très nombreuses unités à disquettes 8" et 5" 1/4 ; ils comprennent un séparateur de données, une horloge pour codage en simple densité (modulation en fréquence) ou double densité (modulation de fréquence modifiée : MFM), un générateur et testeur de redondance cyclique (test d'erreur), un oscillateur, un diviseur et un formatteur logiciel, format IBM 3740 ou système 34, plus des registres comptant les pistes et les secteurs, etc. L'interfaçage d'un processeur et d'une unité à disquettes ne requiert ainsi, outre le 279X, que quelques composants passifs (fig. 5). Elle se fait à partir du circuit de commande d'*accès direct à la mémoire* (DMA, ou ADM) puisque les liaisons procèdent selon ce mode d'échange. Il est évident que l'unité à disquettes comporte, elle aussi, une part d'électronique (fig. 6), avec des portes type 7438 ou 7414 d'interfaçage.

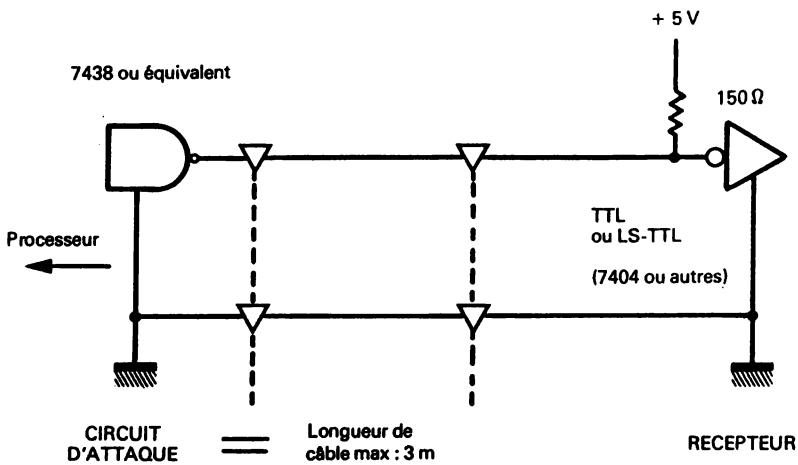


Fig. 7. — Principe de la liaison en sortie du processeur maître.

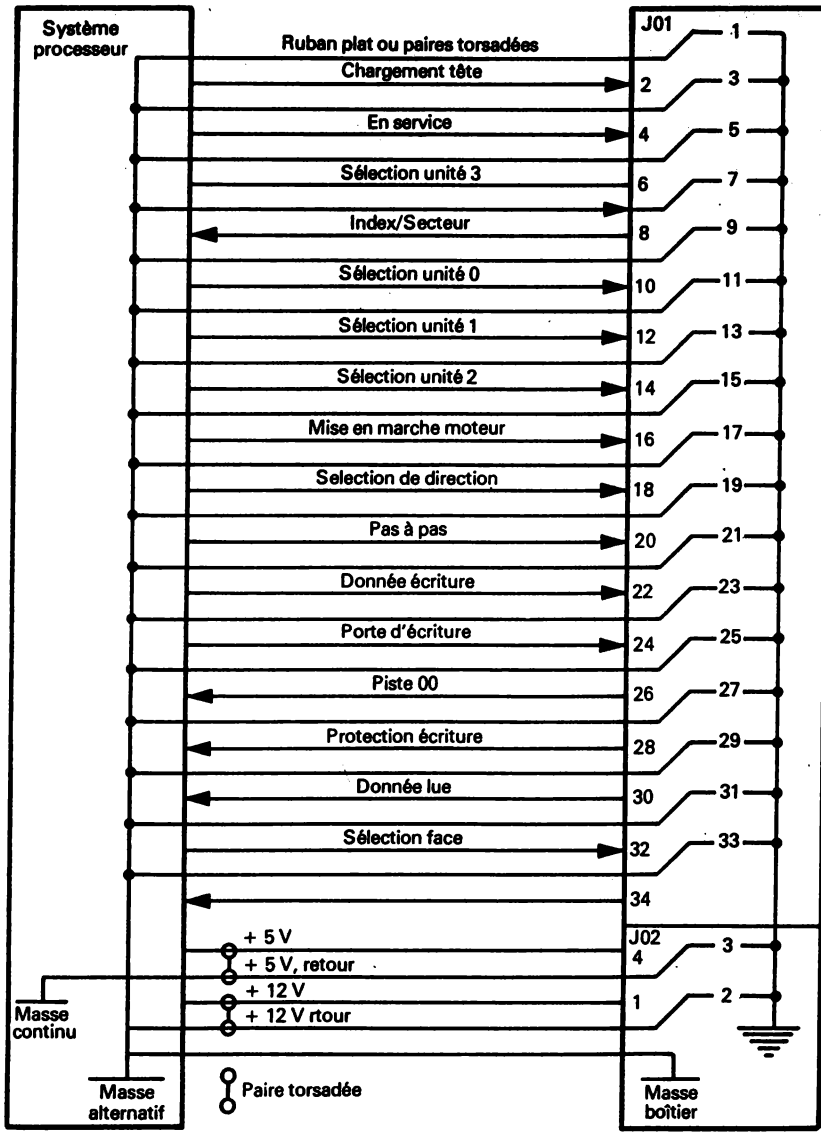


Fig. 8. — Connexion entre le processeur et une unité à microdisquettes 3" Hitachi.

Caractéristiques générales de l'unité à microdisquettes 3" type HFD 305S

Caractéristiques	Densité						Unité
	Simple			Double			
Capacité non formattée							
• par disque	250 K			500 K			Octets
• par surface	125 K			250 K			Octets
• par piste	3125			6250			Octets
Capacité formattée							
• octets par secteur	128	256	512	128	256	512	
• secteurs par piste	16	9	5	16	9	5	
• par disque	164 K	184 K	205 K	328 K	369 K	410 K	Octets
• par surface	82 K	92 K	102 K	164 K	184 K	205 K	Octets
• par piste	2048	2304	2560	4096	4608	5120	Octets
Disquettes							
• nombre de faces	2			2			
• cyclindres	40			40			
• pistes	80			80			
Codage							
• méthode	FM			MFM			
• densité max. :							
- bits par pouce	4473			8946			Bpi
- pistes par pouce	100			100			Tpi
• Vitesse de transfert	125 K			250 K			Bits/s
Temps							
• accès moyen	55						ms
• piste à piste	3						ms
• établissement	15						ms
• attente moyenne	100						ms
Vitesse de rotation (tours par minute)	300						tpm
Erreurs							
• logicielles	10^{-9}						
• matérielles	10^{-12}						

4.2. Interface pour microdisquettes 3 pouces

En 1982, *Hitachi* et *Matsushita* ont annoncé des microdisquettes de trois pouces de diamètre stockant 250 kilo-octets par face, et compatibles avec les unités 5''. Ces disquettes sont contenues dans une enveloppe rigide avec diverses protections, l'unité *Hitachi* contenant déjà son propre microprocesseur.

La liaison avec cette unité « HFD 305S » comporte 11 lignes d'entrées, câblées de préférence comme le montre la figure 7 ; le câble est de 3 m maximum et les niveaux logiques sont de 0 à 0,4 V (zéro) et 2,4 à 5,25 V (un). Les lignes de sortie sont au nombre de 4 (fig. 8). Les caractéristiques générales de cette unité sont résumées dans le tableau.

Ajoutons que *Sony* a, pour sa part, conçu une interface différente pour ses microdisquettes 3''5.

CHAPITRE IX

INTERFACES AVEC LES MONITEURS

Plusieurs techniques d'interfaçage prévalent avec les moniteurs : vidéo composite, RVB, Péritel, VHF ou UHF. Nous allons les examiner ici.

Les mots-clés de ce chapitre

Signaux pour moniteurs : principes, 170
Vidéo composite : ce que c'est, 170
Normes TV, 170

Bande passante : des moniteurs, 171
Vidéo monochrome sur IBM-PC : description, 171
Graphique-couleur sur IBM-PC : description, 174

Péritel : prise pour péri-télévision, 176
Haut-parleur de l'IBM-PC : son connecteur, 178

1. LES PRINCIPES DE COMMANDE D'UN MONITEUR

SIGNAUX POUR MONITEURS

Afin de bien comprendre à quoi correspondent les diverses techniques d'interfaçage d'un moniteur ou d'un écran, il faut se souvenir que leur commande exige la présence de signaux :

- de luminance (suffisant pour un affichage monochrome) ;
- de chrominance : ce sont les signaux de couleur ; ils n'interviennent qu'avec un moniteur couleur ;

La technologie additive utilisée en télévision couleur a fait choisir le rouge, le vert et le bleu comme fondamentales. De ce fait, on parle de « RVB », ou de « RGB » en anglais (red, green, blue).

- de signaux de synchronisation images et lignes.

VIDEO COMPOSITE

Tous ces signaux peuvent être fournis séparément sur un connecteur : c'est ce qu'on appelle une sortie RVB, ou RGB. Ou alors, ils peuvent être combinés ; la sortie se fait sur coaxial et l'on traite de *vidéo composite.*

Le son éventuel d'accompagnement n'a pas été prévu dans ces liaisons, car on suppose qu'on emploie un moniteur professionnel et non un classique récepteur de télévision. Dans ce cas, le haut-parleur est situé à l'intérieur de l'unité centrale. Si l'on voulait disposer du son en plus, il faudrait recourir à une sortie supplémentaire.

Par contre, la liaison Péritel inclut la RVB et la vidéo composite avec une commande de commutation, ainsi que le son en stéréo. Dans ce cas, on utilisera un récepteur de télévision couleur.

NORMES TV

Enfin, certains micro-ordinateurs grand public utilisent les signaux vidéo, synchronisation et son, pour moduler une onde porteuse VHF ou UHF, telle que la reçoit l'antenne d'un téléviseur. Dans un tel cas, la liaison micro-ordinateur à téléviseur s'effectue à l'aide d'un coaxial vers l'entrée antenne du téléviseur, celui-ci devant être calé sur la fréquence de réception correspondante.

Attention, toutefois, aux normes de télévision appliquées : NTSC, PAL ou SECAM, ou même spécifiques, car elles sont incompatibles.

BANDE PASSANTE

Du point de vue de la définition, une caractéristique électronique importante distingue les *moniteurs* des *téléviseurs* : la bande passante. Afin de pouvoir afficher des lignes de 80 caractères, par exemple, il faut pouvoir disposer d'une bande passante de l'ordre de 5 à 20 MHz, caractéristique qu'offrent les moniteurs. Or, celle d'un récepteur de télévision est limitée à bien moins de 5 MHz, ce qui explique que leur affichage ne portera que sur des lignes de 40 caractères seulement pour rester lisible et de très bonne qualité.

2. LA SORTIE VIDÉO MONOCHROME SUR L'IBM-PC

VIDEO MONOCHROME SUR IBM-PC

L'IBM-PC peut piloter d'origine un moniteur monochrome en mode texte uniquement et, avec une carte additionnelle *couleur-graphique*, un écran monochrome ou couleur en modes graphiques. La différence en mode texte est subtile :

• En mode texte, monochrome de base, chaque caractère est défini dans une matrice de 9 par 14 points. L'affichage est excellent.

• Avec la carte graphique-couleur, chaque caractère est défini dans une matrice de 8 par 8 points et, de ce fait, la qualité de l'*affichage des caractères* est moindre.

L'interface moniteur en version monochrome, mode texte, de base, est conçue autour du circuit contrôleur d'écran cathodique type 6845 de *Motorola*. La carte comporte 4 K octets de mémoire vive statique à double accès : par la carte et par l'unité centrale ; aucune parité n'est prévue. Son synoptique simplifié est donné figure 1 :

• l'affichage porte sur 25 lignes de 80 caractères ;
• la matrice de définition est de 9 × 14 points ;
• le caractère lui-même est de 7 × 9 points ;
• le balayage écran se fait à 18 kHz.

Cet adaptateur supporte les 256 caractères, ASCII et définis par IBM, contenus dans un générateur de caractères de 8 K. Les fonctions de clignotement et de double intensité sont disponibles, ainsi que l'inversion vidéo et le soulignement. Notons que l'adresse en mémoire centrale de l'image écran est B0000 en hexadécimal.

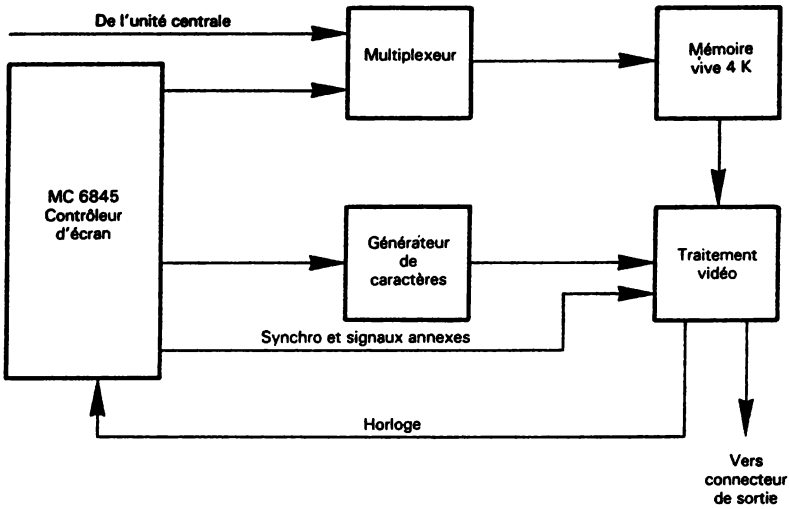
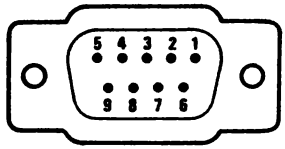


Fig. 1 — Synoptique de principe de l'adaptateur d'affichage monochrome IBM.

Le moniteur courant, de 12 pouces de diagonale (environ 30 cm), offre une bande passante de 16,27 MHz. La définition est de 350 lignes horizontales et 720 traits verticaux, ce qui correspond bien à un affichage de 25 lignes de 80 caractères selon des matrices de 9 par 14 : en effet, 80 fois 9 = 720 et 25 fois 14 = 350. Cinquante images par seconde sont affichées, avec une fréquence de balayage horizontale de 18,432 kHz.

Le connecteur de liaison est un modèle 9 broches, type D, représenté dans la figure 2. Les 3 lignes inutilisées serviront, avec une version couleur, pour les trois composantes RVB. Notons que la même carte adaptateur d'écran monochrome supporte aussi l'interface imprimante.

Le connecteur (a)



Les lignes (b)

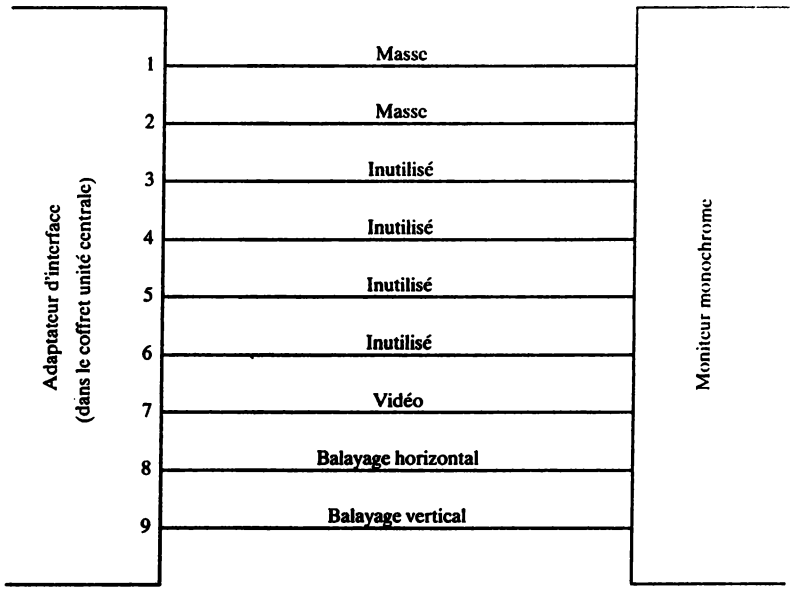


Fig. 2 — Le connecteur femelle 9 broches D, sur la face arrière de l'IBM-PC (en *a*) et les liaisons au moniteur (*b*).

3. LA SORTIE GRAPHIQUE-COULEUR SUR L'IBM-PC

GRAPHIQUE-COULEUR POUR MONITEUR IBM-PC

Une carte d'interface *graphique-couleur* optionnelle peut être introduite dans l'IBM-PC. Elle peut travailler en monochrome ou en couleur et procure des sorties en RVB et en vidéo composite ; elle peut aussi piloter un modulateur HF et recevoir un photostyle (« crayon lumineux »). Ses deux modes de fonctionnement de base sont :

- *l'alphanumérique*, noté A/N ;

- *le graphique*, où tous les points élémentaires sont adressables. Il est noté APA (« All Points Adressable »).

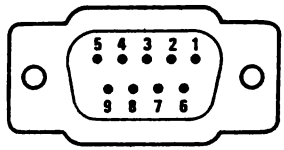
En alphanumérique, l'affichage porte sur 25 lignes de 40 ou 80 caractères, définis dans des matrices de 8×8 points et chacun de 5×7 points. En monochrome, on retrouve la vidéo inversée, le clignotement et la double intensité, mais pas le soulignement (en mode prédéfini). Un choix de 16 couleurs est offert.

Cette interface est également bâtie autour d'un 6845 de *Motorola* et contient 16 K de mémoire d'affichage ; en affichage de 25×80 caractères, 2 K sont employés pour les informations de caractères et deux autres pour les attributs et couleurs, soit 4 K. De ce fait, cette mémoire peut stocker 4 écrans différents (4 « pages »). En mode graphique, la résolution est de 320×200 avec 4 couleurs, ou 640×200 en monochrome, en raison de la capacité mémoire de la carte réduite à 16 K.

La bande passante requise est de 7 MHz en mode texte ou de 14 MHz en graphique. L'adresse en mémoire centrale de l'image écran commence à B8000, en hexadécimal.

Le connecteur femelle d'interface et la liaison sont du même type qu'en monochrome, un connecteur 9 broches type D, mais les 3 lignes auparavant inutilisées sont ici occupées par la RVB (fig. 3). La liaison en vidéo composite se fait sur un classique coaxial, avec des signaux de 1,5 V crête à crête.

Le connecteur (a)



Les lignes (b)

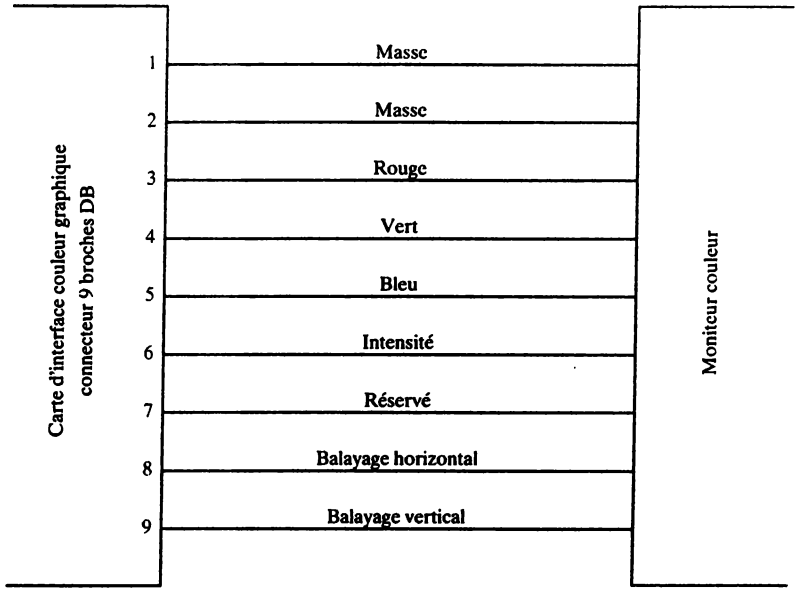


Fig. 3 — Le connecteur 9 broches *(a)* et les lignes de liaison au moniteur couleur de l'IBM-PC *(b)*.

4. LA PRISE PÉRITEL

PERITEL

Parce que le téléviseur est entré dans tous les foyers et parce que son prix est modeste, il peut être aisément utilisé comme terminal pour le télétexte, l'informatique, etc.

« *L'interconnexion entre les dispositifs de péritélévision et les récepteurs de télévision grand public* » a fait l'objet d'une norme NF C 92-250 établie par une commission de l'*UTE* groupant les représentants de l'*EDF*, de la *FIEE (SCART* et *SIERE)*, de l'*AFNOR*, du *CNAM*, du *Comité de Coordination des Télécommunications* (TDF, CCETT, ministère des PTT), du *LCIE* et de la *Diéli* (ministère de l'Industrie) et adoptée le 8 novembre 1978. Elle est disponible à l'AFNOR.

Selon cette norme, le récepteur de télévision domestique devient le terminal visuel universel pour le grand public et devra être adapté aux nouveaux services tels que Antiope, Titan, Epeos, Discret, à la radiodiffusion directe par satellite et à diverses utilisations complémentaires telles que : télésurveillance domestique, jeux électroniques, caméra, magnétoscope, ordinateur familial, etc.

Des caractéristiques d'interconnexion ont donc été définies et elles portent sur les valeurs d'adaptation (tensions et impédances) et sur les caractéristiques dimensionnelles, mécaniques et électriques du connecteur utilisé à ces fins.

La partie *femelle* de ce connecteur à 21 contacts, ou « embase », est montée à demeure sur le récepteur et reliée par câblage aux circuits ou éléments concernés. La partie *mâle*, ou « fiche », est reliée au périphérique intervenant. Les 20 broches sont disposées en deux rangées de 10 contacts en quinconce (plus un *contact spécial de blindage*), au pas de 3,81 mm avec distance entre axes des rangées de 5,08 mm ; leur affectation est définie dans la figure 4.

Les niveaux des signaux et les valeurs d'adaptation sont :

• *En sortie audio, broches 3 et 1* : f.e.m. de 100 mV eff. à ±3 dB, avec une impédance de source au plus égale à 1 kΩ pour les fréquences supérieures à 40 Hz. En monophonie, ces deux broches doivent délivrer des signaux identiques.

• *En entrée audio, broches 6 et 2* : 100 mV eff. ±3 dB, avec une impédance de charge de 10 kΩ et une impédance d'entrée du téléviseur ⩾4,7 kΩ. Ces deux broches sont reliées, en monophonie, de telle sorte que l'impédance entre elles soit au moins de 4,7 kΩ.

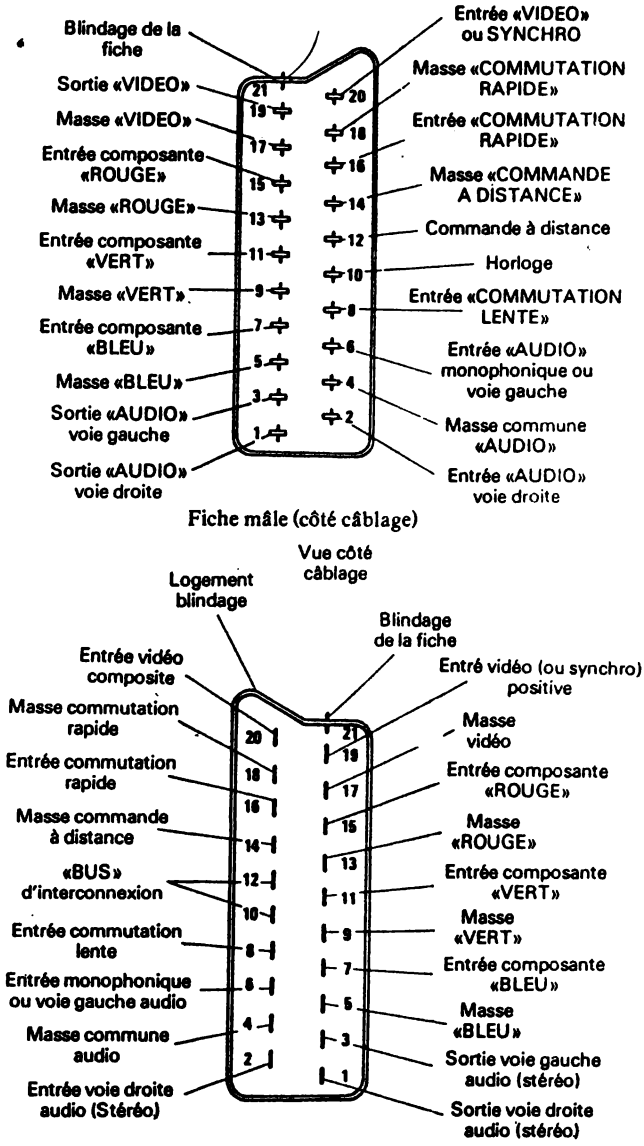


Fig. 4 — La prise Péritel.

5. L'INTERFACE HP DE L'IBM-PC

La prise Péritel comporte des voies « son » ignorées par les liaisons RVB. Dans ce cas, le son fait l'objet d'une liaison distincte. Sur l'IBM-PC, un connecteur spécifique *son* à 4 broches existe sur la carte unité centrale. Le son est piloté par un circuit d'interface parallèle des plus classiques, le 8255, et sort sur un connecteur Berg 4 broches avec les attributions suivantes :

- broche 1 : le son
- broche 2 : clé
- broche 3 : masse
- broche 4 : +5 V.

HAUT-PARLEUR DE L'IBM-PC

Un petit haut-parleur (HP, en abrégé) interne émet les sons programmés.

CHAPITRE X

INTERFACES POUR BANDES MAGNÉTIQUES EN CASSETTES ET CARTOUCHES

En micro-informatique, l'enregistrement d'informations sur cartouches magnétiques sert essentiellement à la sauvegarde des disques durs. Les cassettes, elles, sont utilisées par les machines familiales comme mémoire de masse principale et unique en raison de leur bas prix.

Les mots-clés de ce chapitre

Bandes magnétiques : pour micro-informatique, 180
Cartouche : pour bande magnétique, 180
Cassette : pour bande magnétique, 180

Start-stop : mode pour bande magnétique, 182
Streamer : mode pour bande magnétique, 182
QIC-Ø2 : norme d'interfaçage pour bande quart de pouce, 182

Kansas-City : norme pour cassettes magnétiques, 185
FSK : modulation en fréquence pour norme Kansas-City, 187
Interface cassette sur IBM-PC, 194

1. CARTOUCHES A BANDE QUART DE POUCE

En 1985, les disques durs 5"1/4 courants étaient capables de stocker couramment 10 ou 20 méga-octets. Ces supports étant fixes, inamovibles, la copie de leurs informations pour sauvegarde crée un réel problème. Pour le résoudre, diverses solutions sont apparues :

• *La copie sur disquettes.* Il faut environ 30 disquettes de 360 K (IBM-PC) pour dupliquer un disque dur de 10 Moctets, avec toutes les manipulations que cela suppose. Ce qui décourage bon nombre d'utilisateurs.

• *La copie sur un autre disque dur.* Formule chère mais rapide. Mais que faire si l'on organise une rotation de copies sur 3 jours, par exemple ? Faudra-t-il 3 unités à disque dur ? Ou encore, si l'on veut mettre la copie en sécurité dans un autre lieu ?

BANDES MAGNETIQUES

• *La copie sur bande magnétique en cartouche,* à défilement séquentiel.

• *La copie sur cartouche « streamer »,* à défilement continu.

Dans ces deux derniers cas, la cartouche est amovible, peu onéreuse, et éventuellement, une même unité à cartouches peut desservir plusieurs micro-ordinateurs. Pour des capacités de 10 à 40 méga-octets, telles qu'on les rencontre en micro-informatique, la largeur de bande est généralement de 1/4 de pouce, soit 6,35 mm.

CARTOUCHE

CASSETTE

La cartouche elle-même a pu être normalisée (référence ANSI/ECMA 16). La structure d'une cartouche de référence due à *3M* est ainsi donnée dans la figure 1 ; la longueur de la bande est de 300 pieds (91 m), ou 450 pieds (137 m), ou encore 600 pieds (183 m) ; la capacité peut atteindre plusieurs dizaines de méga-octets. Elle dispose de son propre dispositif de transport, une courroie plastique sans fin entraînée par un cabestan, lui-même mu par le galet animé par un moteur d'entraînement situé sur la platine.

C'est d'ailleurs ce qui différencie *une cartouche* d'une *cassette* : on réserve le nom de cartouche à des chargeurs de bande comportant un dispositif interne d'entraînement. La cassette (genre cassette audio, par exemple) n'en possède pas.

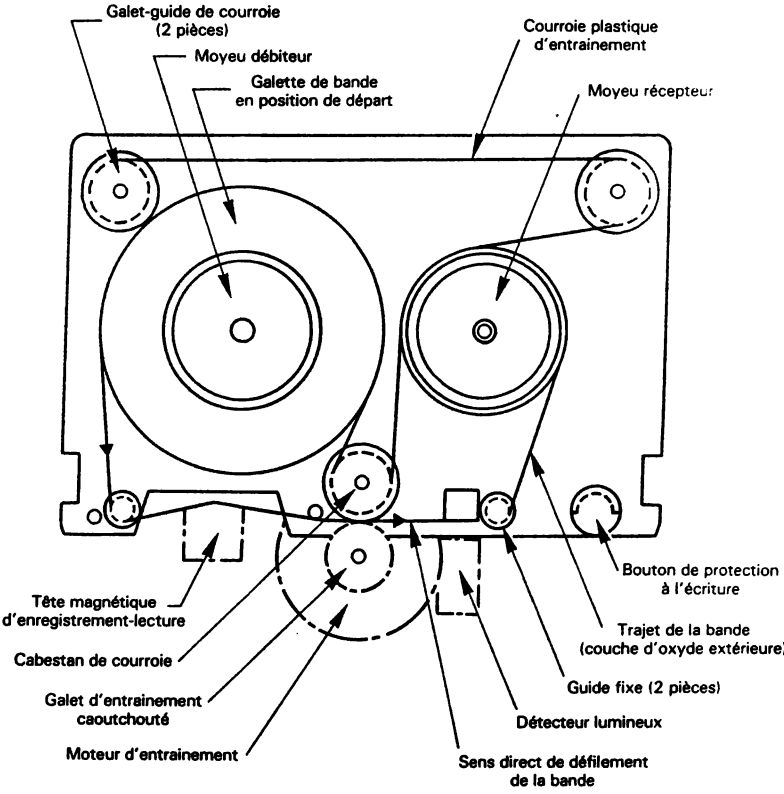


Fig. 1 — Structure de la cartouche 3M, aux normes ANSI/ECMA-16.

Les début et fin de bande sont marqués par des perforations dans la bande, détectées par un dispositif photo-électrique. Le signal obtenu est noté :

• BOT, pour « *Beginning Of Tape* », début de bande, lorsque trois groupes successifs de deux perforations superposées ont été détectées.

• EOT, pour « *End Of Tape* », fin de bande, lors de la détection de trois perforations successives.

2. MODES START-STOP ET STREAMER

La cartouche peut fonctionner selon les modes que lui impose son unité : *en séquentiel*, dit aussi *start-stop*, ou en *défilement continu*, dit aussi en mode *streamer* :

START-STOP (BANDE MAGNETIQUE)
STREAMER

— *En séquentiel* (start-stop), la bande défile mais marque des arrêts et départs. Des espaces sont donc prévus entre deux écritures consécutives afin de permettre ces mouvements ; on les appelle des *espaces inter-blocs* de données, ou IRG (de « *Inter Record Gap* »).

— *En mode streamer*, le défilement est continu, les données sont enregistrées en blocs de longueur fixe (de 512 octets typiquement) placés bout à bout, sans espace ou avec un très court espace entre blocs. De ce fait, et si les données à enregistrer ne se suivent pas, l'unité streamer procède à un repositionnement de la bande.

La capacité en mode streamer est ainsi supérieure, mais il n'y a pas d'avantage évident en temps d'enregistrement pour l'un ou l'autre de ces modes puisque, dans le second cas, tout dépend de la longueur des séries de blocs.

QIC-Ø2

Diverses normes d'interfaçage ont été définies, parmi lesquelles la QIC-Ø2 (QIC provient de « *Quarter Inch Cartridge* », cartouche quart de pouce) pour les streamer, utilisant un protocole d'échanges asynchrones, en poignée de main, et un bus bidirectionnel avec données sur 8 bits (fig. 2). La norme QIC-24 définit le format d'enregistrement (fig. 3) et la position des pistes sur la bande. La vérification des erreurs est assurée par la méthode du CRC (« *Cyclic Redundancy Checksum* »). L'enregistrement, effectué à l'origine sur 4 pistes en « serpentin », par aller et retour, est passé à 9 pistes. Il est encadré par des signaux de BOT (« *Beginning Of Track* », début de piste) et de EOT (« *End Of Track* », fin de piste).

Une autre norme applicable relève de la SCSI telle qu'elle a été adoptée pour les diques.

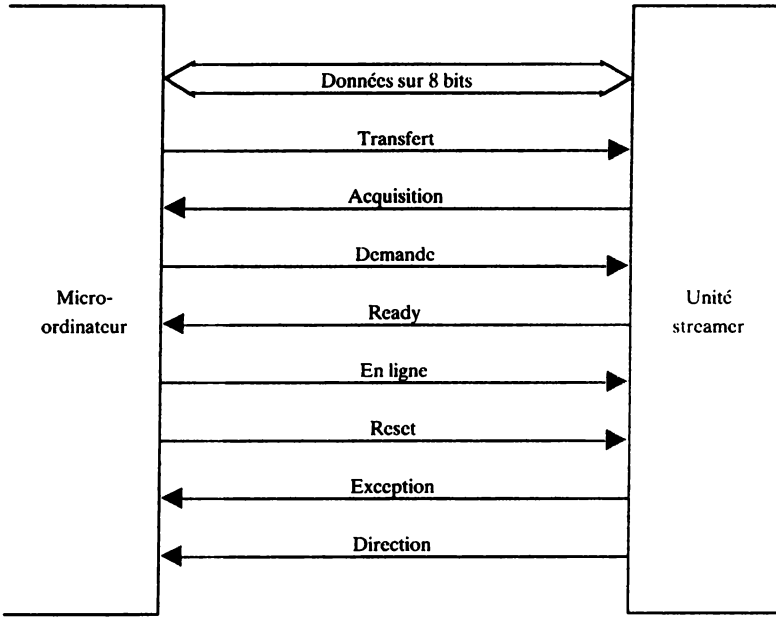


Fig. 2 — Liaison cartouche 1/4 de pouce à unité centrale, selon la norme QIC-02.

Préambule	Marqueur	Données	Adresse du bloc	CRC	Postambule
12 octets minimum	1 octet	512 octets	4 octets	2 octets	4 octets minimum

Fig. 3 — Format d'enregistrement sur bande 1/4 de pouce, selon la norme QIC-24.

3. UTILISATION DE CASSETTES

La cassette, et surtout le classique magnétophone, sont des éléments peu coûteux, simples à utiliser, mais guère rapides ni fiables. Aussi ont-ils surtout été adoptés comme mémoire de masse avec les micro-ordinateurs familiaux.

La structure d'une cassette est bien connue (fig. 4). Selon les modes d'enregistrement, une cassette dite « 60 mn » pourra stocker quelque chose comme 100 K octets. Sa vitesse est réduite car c'est celle du magnétophone. Notons, à ce propos, que certains fabricants ont imaginé des unités professionnelles utilisant des cassettes « numériques » de hautes qualités, fonctionnant à plusieurs vitesses (bobinage, recherche, lecture ou écriture). Nous allons examiner, ici, le seul cas des matériels courants.

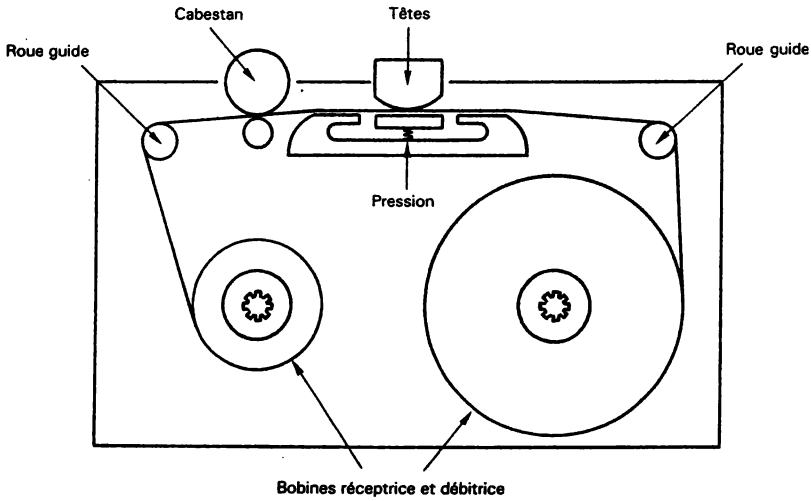


Fig. 4 — Cassette de bande magnétique courante.

L'enregistrement s'effectue en mode série, obligatoirement, et applique donc les principes que nous avons décrit avec les liaisons *séries asynchrones* : les caractères, par octets, sont encadrés de bits de *start* et de *stop*. En outre, et afin d'enregistrer des impulsions binaires sur bande magnétique audio, il faut

les transformer en ondes audibles ; en effet, les magnétophones n'acceptent pas les fréquences binaires traitées par le micro-ordinateur ; en outre, il faut tenir compte du fait qu'un train de *zéros* ou de 1 binaires correspond à une composante continue, non acceptée par l'électronique des magnétophones.

KANSAS CITY

Divers modes de traduction binaire à fréquences audibles ont été proposés. La norme la plus courante est la « Kansas City » ; c'est en effet dans cette ville des Etats-Unis qu'en novembre 1975, un symposium, organisé par la revue *Byte*, contribuait à la définir. Selon cette norme :

- le 0 binaire est traduit par 4 cycles d'une onde à 1 200 Hz ;
- le 1 binaire est traduit par 8 cycles d'une onde de 2 400 Hz ;
- un caractère est enregistré en mode série asynchrone avec *un bit de start, 8 bits de donnée, 2 bits de stop* au moins (soit 11 bits au moins) ;
- l'intervalle entre caractères peut être quelconque ;
- le bit de plus faible poids (LSB) est transmis en premier, celui de plus fort poids (MSB) en dernier ;
- les données sont enregistrées en *blocs* de longueur variable, arbitraire et optionnelle ;
- un bloc est précédé et suivi par au moins 5 secondes de niveaux 1 consécutifs ;
- les données significatives ne doivent pas être enregistrées sur les 30 premières secondes du début de bande.

La figure 5 montre par quoi se traduisent les 0 et 1 binaires. Notez que leur durée en temps est identique : 1 période à 1 200 Hz dure $1/1200 = 0{,}83$ ms et 4 périodes occupent $0{,}83 \times 4 = 3{,}33$ ms. Il en va de même avec le 2 400 Hz ; une période dure 0,42 ms et 8 occupent aussi 3,33 ms (nous avons arrondi les valeurs).

Par conséquent, on peut transmettre 1000/3,33 = 300 bits par seconde. Soit, avec 11 bits par caractère, 300/11 = 27 caractères maximum par seconde.

On en conclut que la vitesse reste très faible. A la lecture, bien entendu, on procèdera à la reconstitution du signal binaire à partir des 1 200 et 2 400 Hz lus. Par conséquent, l'interface idéale sera constituée par un circuit du type UART *(« Universal Asynchronous Receiver Transmitter »)* chargé de convertir en mode *série* les octets *parallèles* émis par un microprocesseur, pour l'enregistrement sur bande, l'inverse à la lecture.

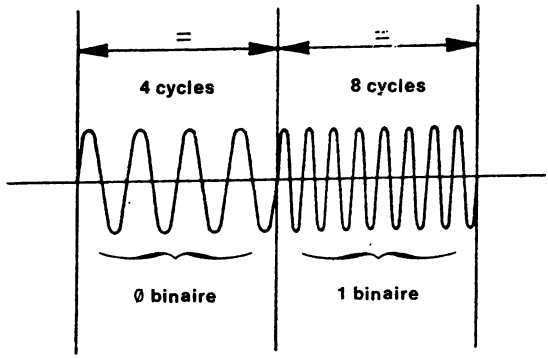


Fig. 5 — Enregistrement selon le standard Kansas City sur cassette magnétique.

Une cassette pourra recevoir plusieurs séquences d'informations, appelées *enregistrements* et séparées par un espace quelconque. Chaque enregistrement comprend à son tour :

1. *Un marquage de départ*, constitué par une série de niveaux *hauts* successifs.

2. *Un code d'identification*, appelé ID, et le nombre d'octets de l'enregistrement. Il est sur 2 octets, généralement. Le nombre d'octets servira à calculer la fin de l'enregistrement.

3. *Les données* elles-mêmes.

4. *Le total* des octets enregistrés, calculé au fur et à mesure par le microprocesseur. Il servira à détecter des erreurs lors d'une relecture : le total enregistré devra correspondre au total des octets lus.

Par sécurité, on enregistrera deux fois successivement les mêmes séquences. Parfois même, trois fois.

FSK

Ce mode d'enregistrement sur une double fréquence est appelé FSK, de « Frequency Shift Keying », soit commutation de fréquence.

On peut, en effet, imaginer qu'on dispose de deux oscillateurs différents que l'on commute selon les besoins.

4. MONTAGES AUX NORMES KANSAS CITY

Le schéma de principe de l'interface circuits est donné par la figure 6. L'UART exécute non seulement la conversion parallèle-série à l'enregistrement mais, fonctionnant en mode asynchrone, ajoute un bit de *start* et deux bits de *stop* à chaque caractère. La parité est rarement utilisée dans la mesure où le *total* calculé des octets intervient dans la détection des erreurs. A l'émission tout comme à la réception, on fait appel à une horloge 4 800 Hz (pouvant d'ailleurs dériver de l'horloge principale, par division par le microprocesseur) ; divisée par 16, elle procure les 300 Hz où chaque période correspond à $1/300 = 3{,}33$ ms, durée que nous avions définie pour marquer un bit.

La société *Motorola* a pratiquement respecté l'essentiel de cette norme sur une carte micro-ordinateur d'évaluation (référence MEK 6800 D2), ajoutant par voie logicielle :

• 1 024 niveaux *hauts* consécutifs en début d'enregistrement, afin de laisser la bande défiler environ 30 s ;

• la lettre B, pour marquer le *début de bande* (le « BOT » : *Beginning Of Tape*), en code ASCII ;

• un octet indiquant la longueur d'un bloc (256 octets maximum, par conséquent) ;

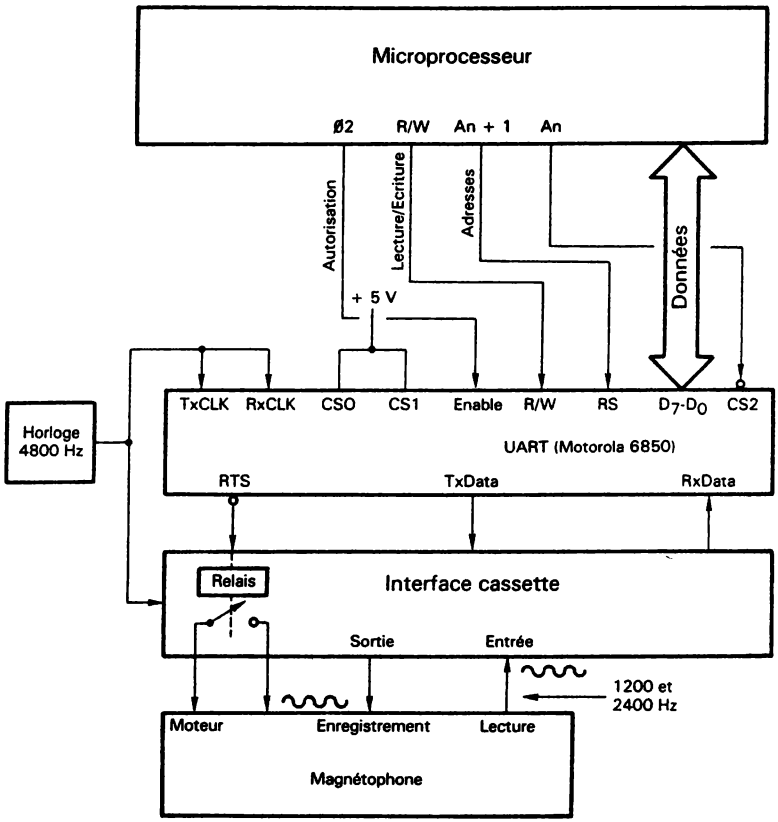


Fig. 6 — Synoptique d'une interface cassette, organisée autour d'un UART (émetteur-récepteur universel asynchrone) exécutant la transposition parallèle à série, et vice-versa.

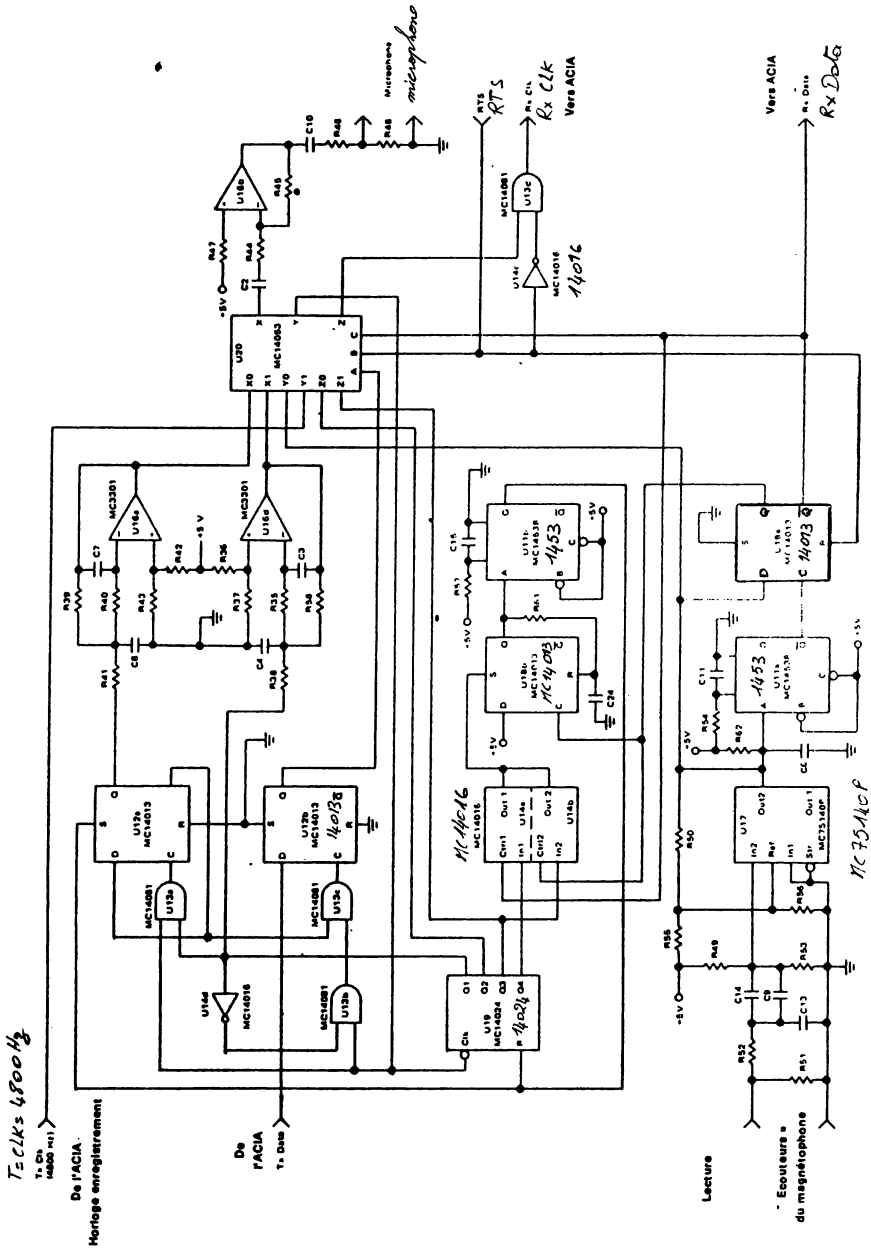


Fig. 7 — Interface pour cassette audio, en Kansas City (source : Motorola).

• deux octets fournissant l'adresse mémoire à partir de laquelle les données à transmettre sont lues ;

• 256 octets maximum de données ;

• 25 niveaux *hauts* en tant qu'espaces ;

• le caractère ASCII « G » pour marquer la fin du bloc.

Le schéma électronique fait appel à un UART appelé par *Motorola* ACIA pour « *Asynchronous Interface Adapter* », circuit référence MC6850 de notre schéma précédent, suivi par le montage de la figure 7 où l'on suppose que le train binaire est déjà reçu complet via la broche *TxData*, pour enregistrement, ou apparaît complet en lecture en *RxData*. Les organigrammes correspondants sont donnés dans les figures 8, 9 et 10.

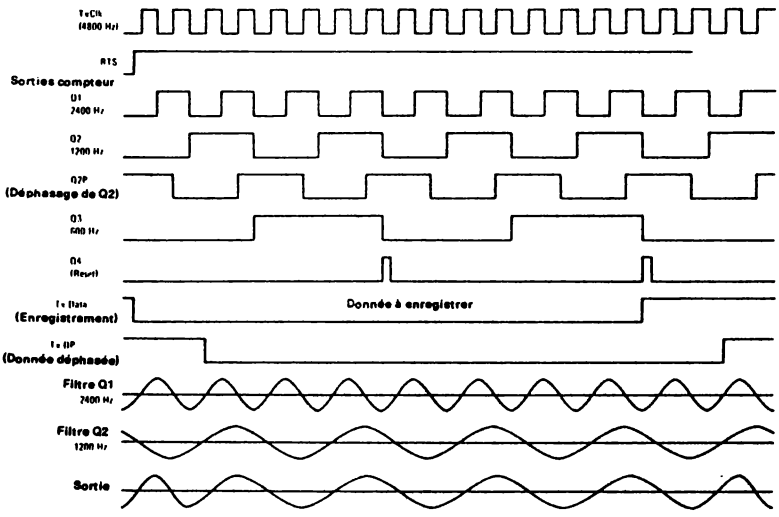


Fig. 8 — Ondes disponibles à l'enregistrement sur le schéma de la figure précédente.

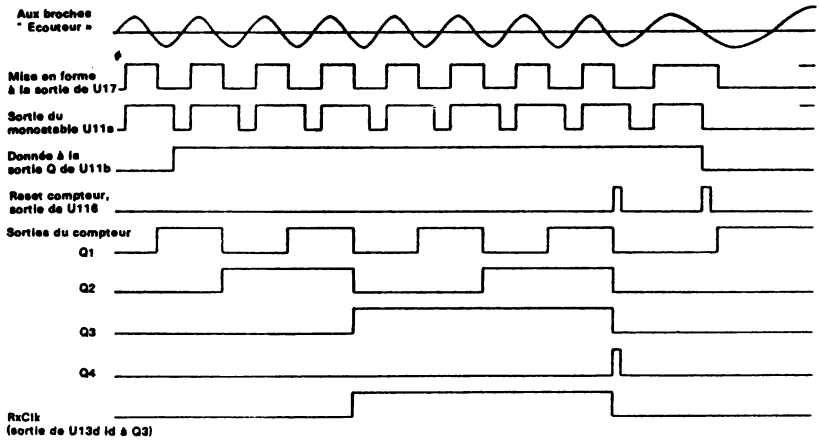


Fig. 9 — **Lecture : transition zéro à 1.**

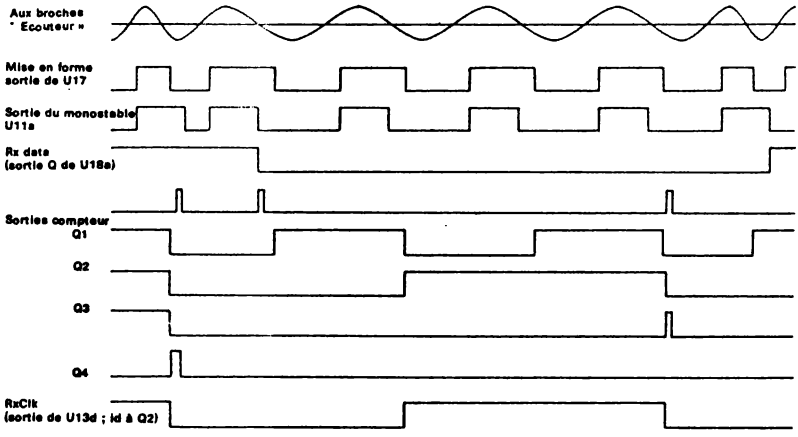


Fig. 10 — Lecture : transition du 1 au zéro.

On trouvera le détail de ce montage dans l'ouvrage *Motorola* « MEK6800D2 Evaluation Kit II Manual ».

Un montage plus simple, comportant un traitement logiciel plus intense, a été proposé par *Motorola* en variante avec le micro-ordinateur d'évaluation MEK6802D5E. Là, les accès série en lecture et écriture sont prélevées sur un circuit banal d'interface parallèle, le PIA (référence MC6821) dont on n'utilise que deux broches sur les 20 disponibles en E/S. Le circuit électronique ne comporte alors plus que des amplificateurs (fig. 11).

Côté magnétophone

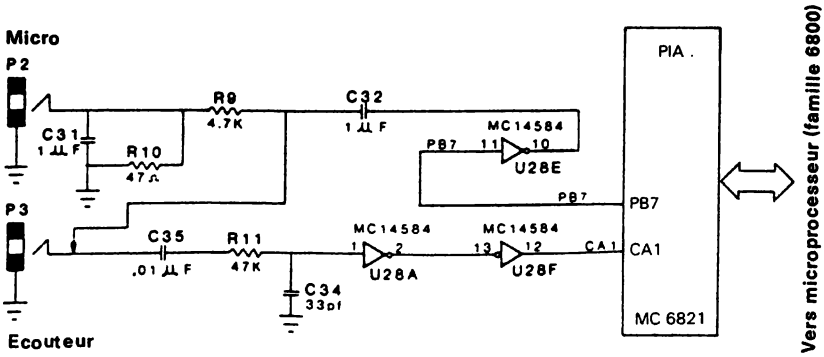


Fig. 11 — Commande de magnétophone au standard Kansas City à partir d'une interface parallèle PIA dont on n'utilise que deux fils. Le traitement est exécuté par le logiciel.

Une méthode encore plus simple a été proposée par *Intel* avec le microprocesseur 8085 puisque celui-ci comporte déjà des lignes séries d'entrée et de sortie, appelées SID et SOD. Le standard n'est toutefois plus respecté car *Intel* divise chaque enregistrement d'un bit en trois temps ; selon qu'il s'agit d'un *zéro* ou d'un 1, une salve d'impulsion sera émise pendant 1 ou 2 temps (fig. 12). Le schéma électronique est ainsi celui de la figure 13, les ondes obtenues étant représentées dans la figure 14.

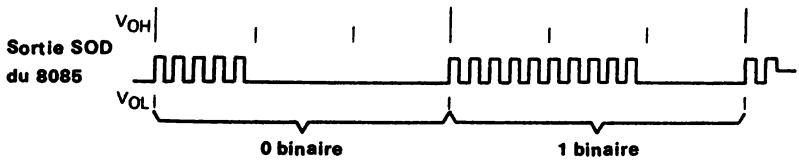


Fig. 12 — Intel a proposé d'enregistrer les 0 et 1 à l'aide de trains d'impulsions occupant 1/3 ou 2/3 du temps imparti à 1 bit.

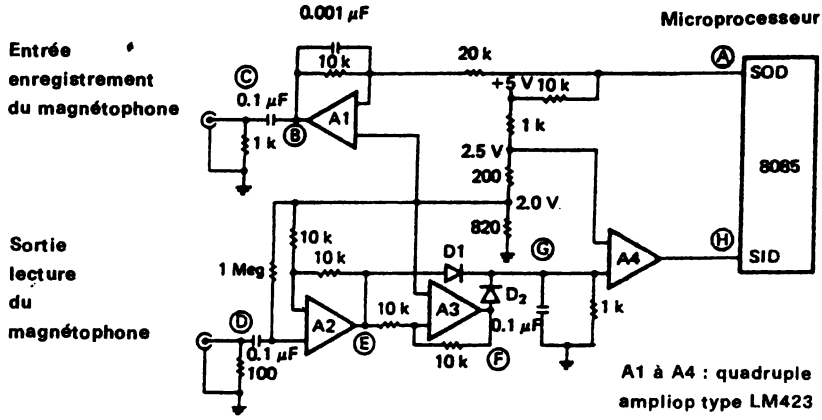


Fig. 13 — Commande d'un magnétophone à partir du microprocesseur 8085, selon un montage et une technique Intel.

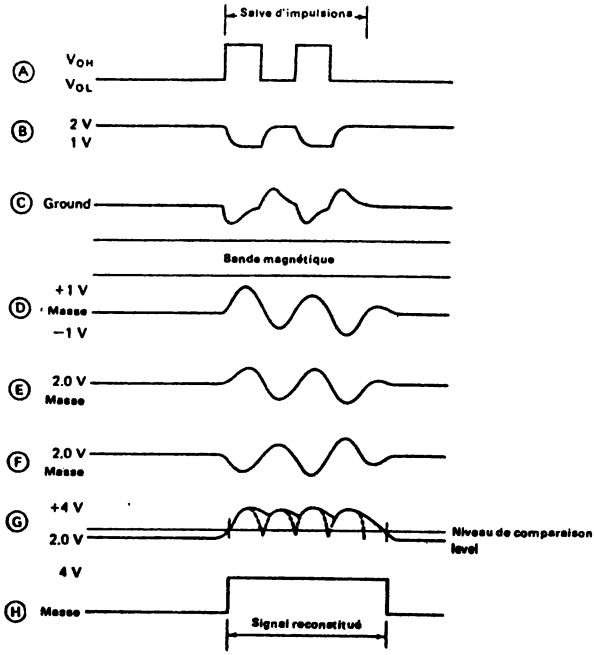


Fig. 14 — Signaux relevés sur le montage précédent.

Ajoutons que le programme de commande en Assembleur a été développé et présenté par M. Rafiquzzaman dans un livre, « Microcomputer Theory and Applications with the Intel SDK-85 » publié en 1982 par *John Wiley & Sons.*

5. INTERFACE CASSETTE DE L'IBM-PC

INTERFACE CASSETTE SUR IBM-PC

Les premières versions de l'IBM-PC, orientées grand public, comportaient une interface cassette supprimée par la suite. C'est un circuit d'interface parallèle classique, le 8255, qui intervient, l'essentiel des fonctions étant géré par logiciel. Le timer est un circuit 8253. Le connecteur de sortie sur l'unité centrale est un DIN à 5 broches (fig. 15).

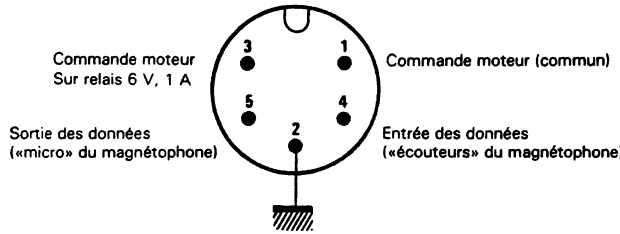


Fig. 15 — Le connecteur d'interface cassette sur la face arrière des premiers IBM-PC.

La commande du moteur s'opère via un relais, lui-même géré par une broche du 8255. Les vitesses sont de 1 000 ou 2 000 bauds et les tensions de sorties, sélectionnables par un cavalier, de 68 mV (accès « Aux » d'un magnétophone) ou de 75 mV (accès « Micro »). La tension d'entrée (lue) est amplifiée par un amplificateur opérationnel type 741.

CHAPITRE XI

LES BUS HP-IB (IEEE-488) ET HP-IL

Le bus HP-IB est également appelé GP-IB ou IEEE-488. Il sert à relier un ordinateur à des périphériques, terminaux informatiques, appareils de mesure les plus divers. Ses applications vont des expériences de laboratoire au contrôle de production, et il a fait l'objet d'une réelle normalisation.

On va l'étudier ici, mais on examinera aussi un concept plus récent, proposé par la même société Hewlett-Packard, qui a mené au mode de liaison série HP-IL.

Les mots-clés de ce chapitre

IEEE-488 : bus normalisé, 196
HP-IB : autre nom du bus IEEE-488, 196
GP-IB : autre nom du bus IEEE-488, 196

IEC 625-1 : norme CEI correspondant au IEEE-488, 196
Circuits 8291 et 8292 : circuits intégrés IEEE-488, 208
HP-IL : bus série de Hewlett-Packard, 209

1. HP-IB ET IEEE-488

Puisque, de plus en plus, les appareils de mesure sont soumis à, ou correspondent avec un ordinateur, il était intéressant de normaliser leur mode d'interconnexion. C'est ce qu'a fait la société *Hewlett-Packard* dès 1973-1974 en créant un bus « HP-IB » *(« Hewlett-Packard Interface Bus »)*. Le HP-IB définit ainsi un nombre de fils, la nature des signaux qui y circulent, leurs fonctions et leur séquencement, l'organisation du connecteur et, plus généralement, les règles d'échanges entre les divers matériels appelés à dialoguer.

Ainsi, actuellement, nombreux sont les appareils de mesure, mais aussi des périphériques, a être dotés du bus HP-IB : générateurs, multimètres, analyseurs, alimentations, imprimantes, etc. Grâce à ce bus, la réalisation d'un ensemble de mesure piloté par ordinateur se réduit à une simple connexion.

IEEE-488
HP-IB
GP-IB
IEC.625-1

En raison de son intérêt, le bus HP-IB a ensuite été repris puis normalisé par l'organisation américaine *IEEE (« Institute of Electrical and Electronics Engineers »)* sous le nom de IEEE-488, suivi de l'année : 1975 pour la première norme. En effet, des compléments lui ont été apportés par la suite, et il a été révisé en 1978. Sa dénomination exacte est alors devenue « *IEEE Standard Digital Interface for Programmable Instrumentation* », mais il est également connu sous les noms de :

- GP-IB, pour « *General Purpose Interface Bus* », soit *bus d'interface à usage général.*

- IEC 625-1, norme de l'*International Electrical Commission*, ou CEI (mais avec un connecteur sensiblement différent, ce qui n'est pas fait pour simplifier les choses !).

2. CARACTÉRISTIQUES ET STRUCTURE

Si l'on suppose que des ordinateurs, surtout des micro-ordinateurs, ainsi que des périphériques, et des appareils de mesure sont dotés d'une sortie IEEE-488, on pourra les interconnecter directement, ce qui simplifie considérablement un problème qu'on avait auparavant beaucoup de difficultés à résoudre. Il existe cependant certaines restrictions qu'il convient d'indiquer :

• Le nombre d'appareils connectés est de 15 au maximum.

• Plus de la moitié doit être effectivement en marche.

• Deux appareils quelconques ne doivent pas être distants de plus de quatre mètres.

• La longueur du câble de liaison ne doit pas dépasser 2 m entre appareils pour une longueur totale de 20 mètres.

• La vitesse maximale de transfert est de 1 méga-octet par seconde (pour un câble de 10 m ou moins et des appareils distants de 1 m). Pour des longueurs supérieures, la vitesse sera moindre, par exemple 500 000 ou 250 000 octets par seconde. Il est important de noter que la vitesse de transmission est *celle de l'appareil le plus lent* (échanges en asynchrone), au cours d'un échange précis, bien entendu ; cela signifie qu'un appareil lent *non adressé* ne réduit pas la cadence de transfert. Les échanges sont donc assez lents.

• Le contenu des messages échangés entre les appareils n'est absolument pas normalisé.

En effet, le standard assure les compatibilités mécanique et électrique, et une synchronisation correcte.

L'architecture classique du bus est représenté dans la figure 1. Le bus comprend 16 lignes se divisant en :

• 8 pour les données (par octet) ;
• 8 pour le contrôle et la gestion des échanges.

Dans le schéma, on voit que le bus est connecté à des appareils qui peuvent être :

— *un contrôleur* (c'est l'unité centrale, l'ordinateur) ;

— *des appareils « auditeurs »* qui interviennent uniquement en recevant des informations ;

— *des appareils « orateurs »* qui interviennent en émettant des informations mais qui peuvent recevoir des ordres ;

— *des appareils à la fois auditeurs et orateurs.*

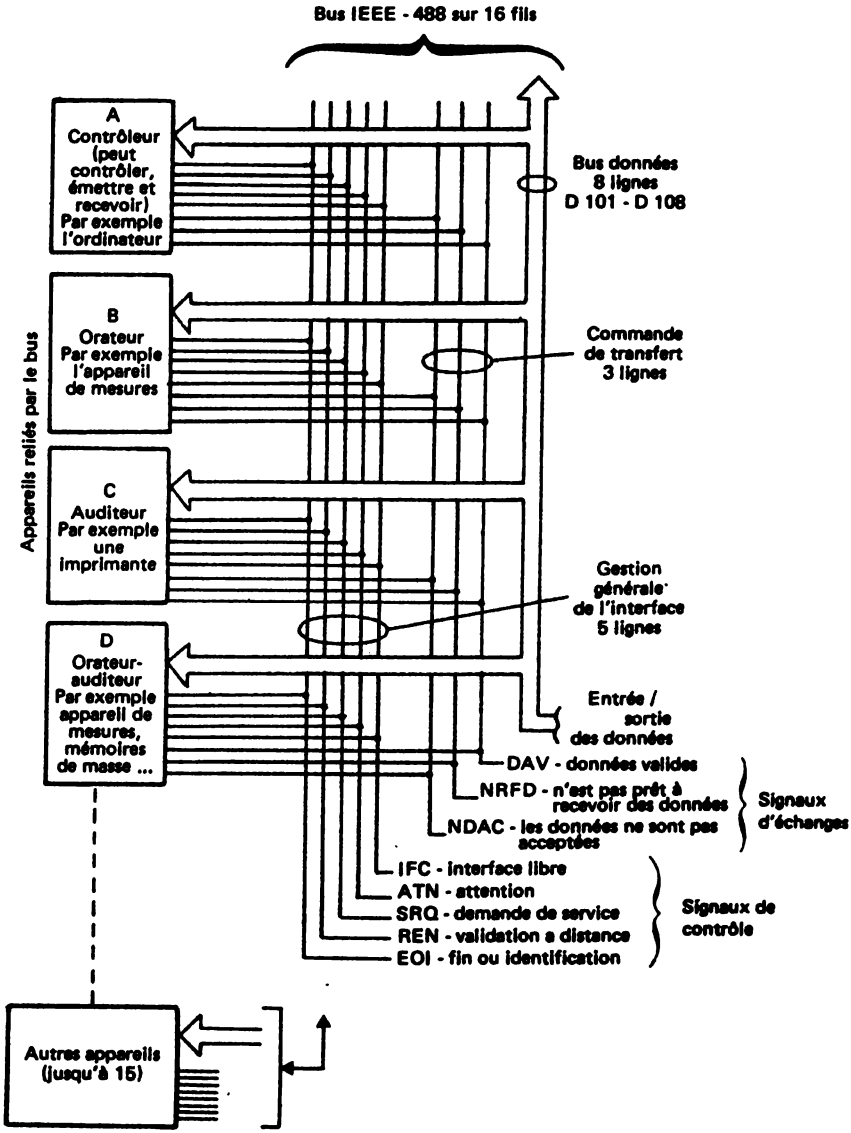

Bus IEEE - 488 sur 16 fils
A
Contrôleur (peut contrôler, émettre et recevoir) Par exemple l'ordinateur
B
Orateur Par exemple l'appareil de mesures
C
Auditeur Par exemple une imprimante
D
Orateur-auditeur Par exemple appareil de mesures, mémoires de masse ...
Autres appareils (jusqu'à 15)
Appareils reliés par le bus
Bus données 8 lignes D 101 - D 108
Commande de transfert 3 lignes
Gestion générale de l'interface 5 lignes
Entrée / sortie des données
DAV - données valides
NRFD - n'est pas prêt à recevoir des données
NDAC - les données ne sont pas acceptées
Signaux d'échanges
IFC - interface libre
ATN - attention
SRQ - demande de service
REN - validation a distance
EOI - fin ou identification
Signaux de contrôle

Fig. 1 — Le bus normalisé IEEE-488 ou HPIB, ou encore GPIB, a été typiquement conçu pour relier des appareils d'instrumentation à un contrôleur.

AUDITEUR
ORATEUR

Ces vocables d'*auditeur* et d'*orateur* traduisent les termes de *listener* et de *talker* adoptés par les Américains. Plusieurs auditeurs peuvent fonctionner simultanément, mais à un instant quelconque, un seul orateur est actif. Dans une chaîne minimale comprenant un voltmètre numérique et une imprimante, c'est évidemment le voltmètre qui sera l'orateur et l'imprimante l'auditeur.

Si la chaîne est plus complexe, le contrôleur dirigera les échanges en activant les appareils et en contrôlant le processus en temps réel. S'il y a plusieurs contrôleurs, un seul est actif à un instant donné. Cette fonction peut être transférée à l'un des appareils du système. Une autre particularité, c'est que les appareils peuvent converser entre eux sans intervention du contrôleur *une fois que celui-ci a désigné l'émetteur et les récepteurs*.

Enfin, le connecteur du HP-IB est à 24 broches ; il est dessiné figure 2 avec ses affectations. Les symboles utilisés pour les commandes sont donnés ci-après.

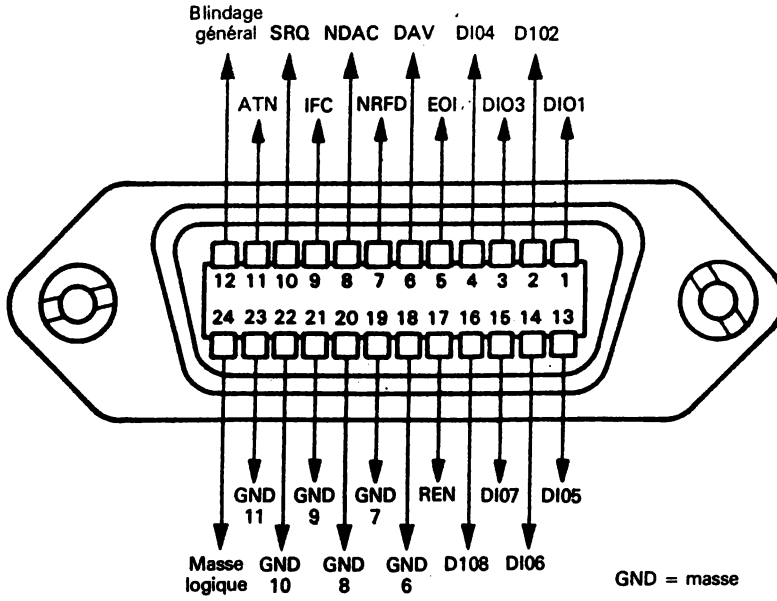


Fig. 2 — Le connecteur du bus IEEE-488.

3. DESCRIPTION MÉCANIQUE

Le câble de liaison constitue le bus proprement dit. C'est un câble passif, par lequel les informations transitent dans les deux sens. Il comporte 16 lignes, une masse logique, cinq blindages partiels et un blindage général, et se termine par des prises mâle/femelle à chaque extrémité. Ses 24 lignes se répartissent en :

1) *Bus de données*, avec 8 lignes numérotées DIO 1 à DIO 8 (où DIO signifie « *data in-out* » : entrée-sortie des données). Les données transitent donc sur un octet et peuvent être numériques, alphanumériques, binaires, des adresses de périphériques, des commandes normalisées, ou des mots d'état.

2) *Bus d'échanges* « en poignée de main » ; il est composé de trois lignes :
— DAV, pour « *Data Valid* », donnée valide ;
— NRFD, pour « *Not Ready For Data* », non prêt pour les données ;
— NDAC, pour « *Not Data Accepted* », donnée non acceptée.

3) *Bus de commande*, avec ses cinq lignes :
— ATN, pour *attention* ;
— REN, pour « *Remote Enable* », autorisation de télécommande ;
— IFC, pour « *Interface Clear* », interface remise à zéro ;
— EOI, pour « *End Or Identify* », fin ou identification ;
— SRQ, pour « *Service Request* », demande de service.

Les trois lignes du bus d'échanges, DAV, NRFD et NDAC, ainsi que les lignes ATN, IFC et SRQ, disposent de leur masse logique individuelle. Il s'y ajoute la masse logique générale et le blindage général. Le nombre élevé des fils de masse s'explique par la nécessité d'éviter les interactions électriques entre les principaux fils de contrôle, chacun de ceux-ci étant torsadé avec son fil de masse.

Le connecteur monté sur chaque appareil est un connecteur femelle. Le câble de liaison, de 0,5 m à 4 m de longueur, est garni à chaque extrémité d'un connecteur mâle/femelle et, de ce fait, le câble connecté laisse apparaître une nouvelle prise femelle (fig. 3). Le bus peut ainsi être organisé en guirlande, en étoile ou en combinaison de ces deux méthodes (fig. 4).

Chaque appareil doit disposer, de préférence à proximité du connecteur, d'un jeu de cinq micro-interrupteurs positionnés de façon à créer une combinaison unique caractérisant l'*adresse* de l'appareil. Il s'y ajoutera un inverseur de sélection de mode « Listen only » *(auditeur seulement)* ou « Talk only » *(orateur seulement)*, avec une seconde position « Adressable ».

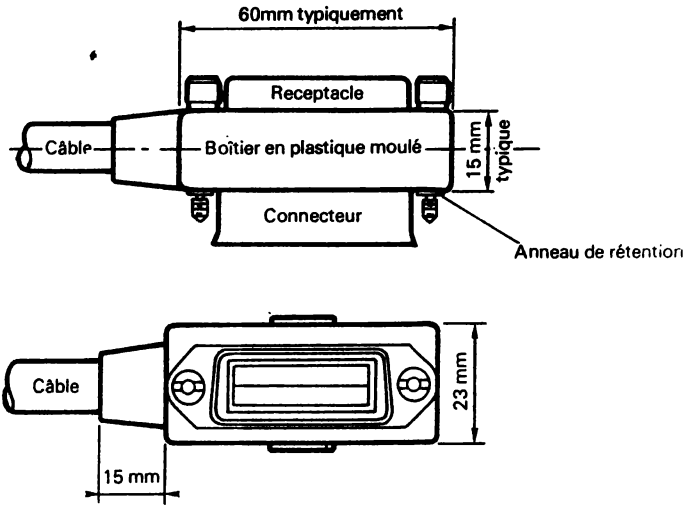
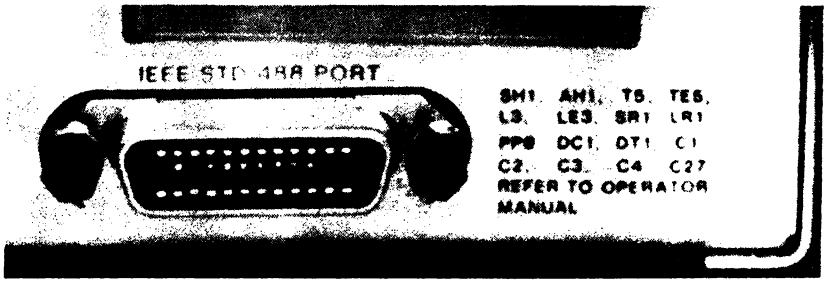


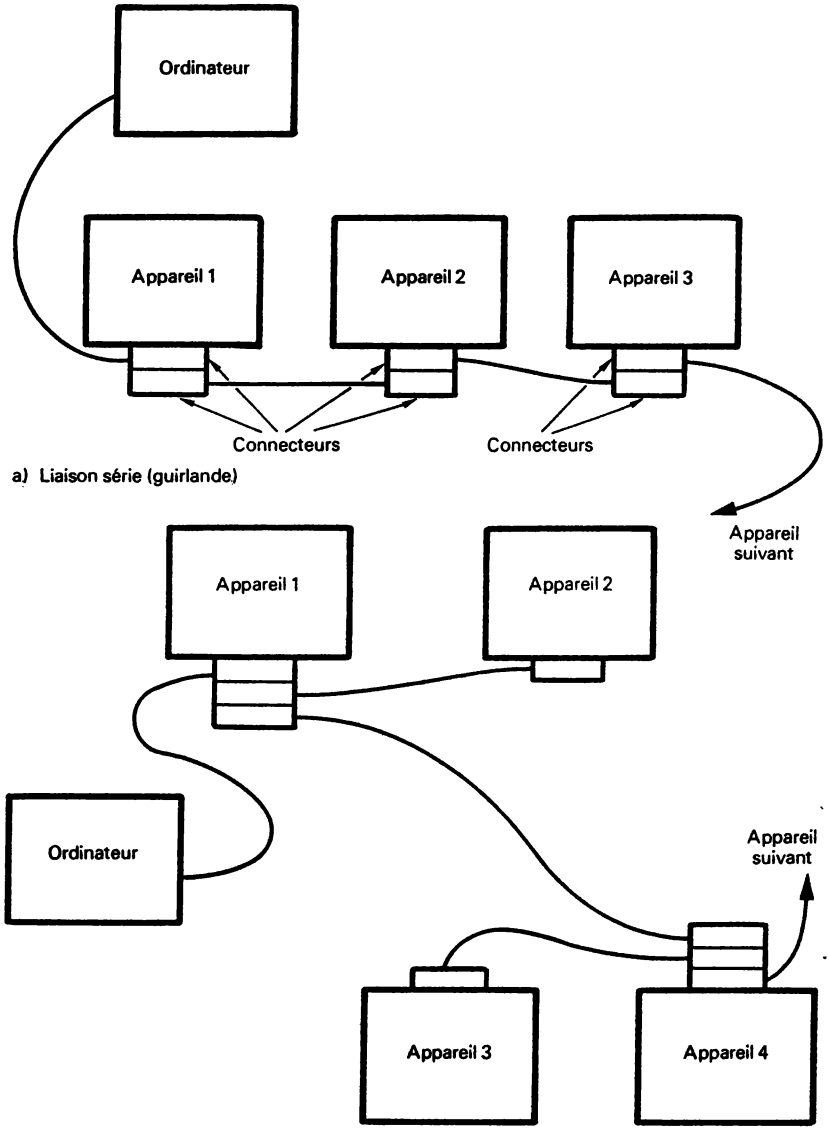
Fig. 3 — Le connecteur mâle/femelle, qui équipe chaque extrémité du câble type IEEE-488.

Ajoutons que le connecteur IEC compte 25 broches, alors que le IEEE standard n'en comporte que 24 ; d'autre part, la distribution des fonctions n'est pas semblable.

Les signaux électriques sur les lignes sont compatibles TTL, en logique négative. Ainsi, le niveau zéro va de 2 à 5 V et le niveau 1 est de moins de 0,8 V. Les lignes sont généralement issues d'un transistor driver devant être capable de fournir 48 mA à l'état *bas*. Un schéma-type est donné figure 5 ; la présence des résistances de charge oblige à mettre en service simultanément plus de la moitié des appareils, faute de quoi le niveau *haut* logique ne pourrait atteindre la tension requise.



Le connecteur IEEE-488.



a) Liaison série (guirlande)

b) Montage mixte, série et étoile

Fig. 4 — Les façons de connecter les appareils au bus IEEE-488.

Affectation des broches dans le cas des normes IEEE-488 et IEC-625.

IEEE Standard			IEC Standard		
broche	Désignation		broche	Désignation	
1	DIO1		1	DIO1	
2	DIO2		2	DIO2	
3	DIO3		3	DIO3	
4	DIO4		4	DIO4	
5	EOI		5	REN	
6	DAV		6	EOI	
7	NRFD		7	DAV	
8	NDAC		8	NRFD	
9	IFC		9	NDAC	
10	SRQ		10	IFC	
11	ATN		11	SRQ	
12	SHIELD		12	ATN	
13	DIO5		13	SHILD	
14	DIO6		14	DIO5	
15	DIO7		15	DIO6	
16	DIO8		16	DIO7	
17	REN		17	DIO8	
18	GND 6		18	GND 5	
19	GND 7	broches	19	GND 6	broches
20	GND 8	de terre	20	GND 7	de terre
21	GND 9	pour les lignes	21	GND 8	pour les lignes
22	GND 10	6 à 11 incluse	22	GND 9	5 à 12 incluse
23	GND 11		23	GND 10	
24	LOGIC GND		24	GND 11	
			25	GND 12	

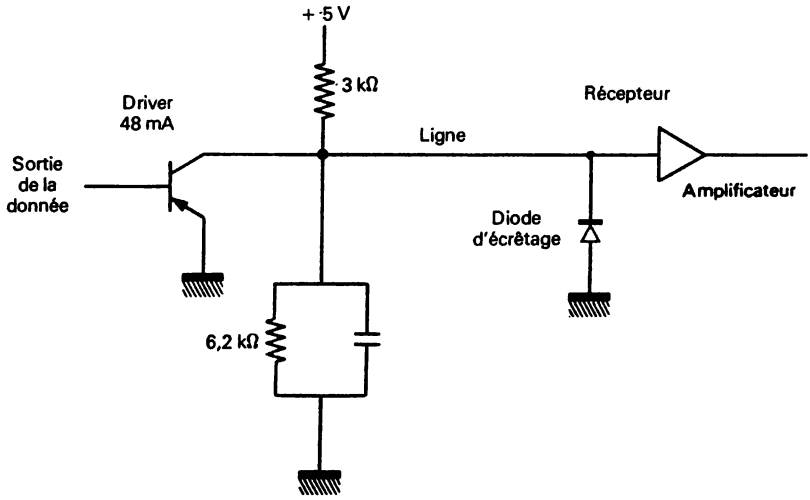


Fig. 5 — Montage de l'attaque d'une ligne.

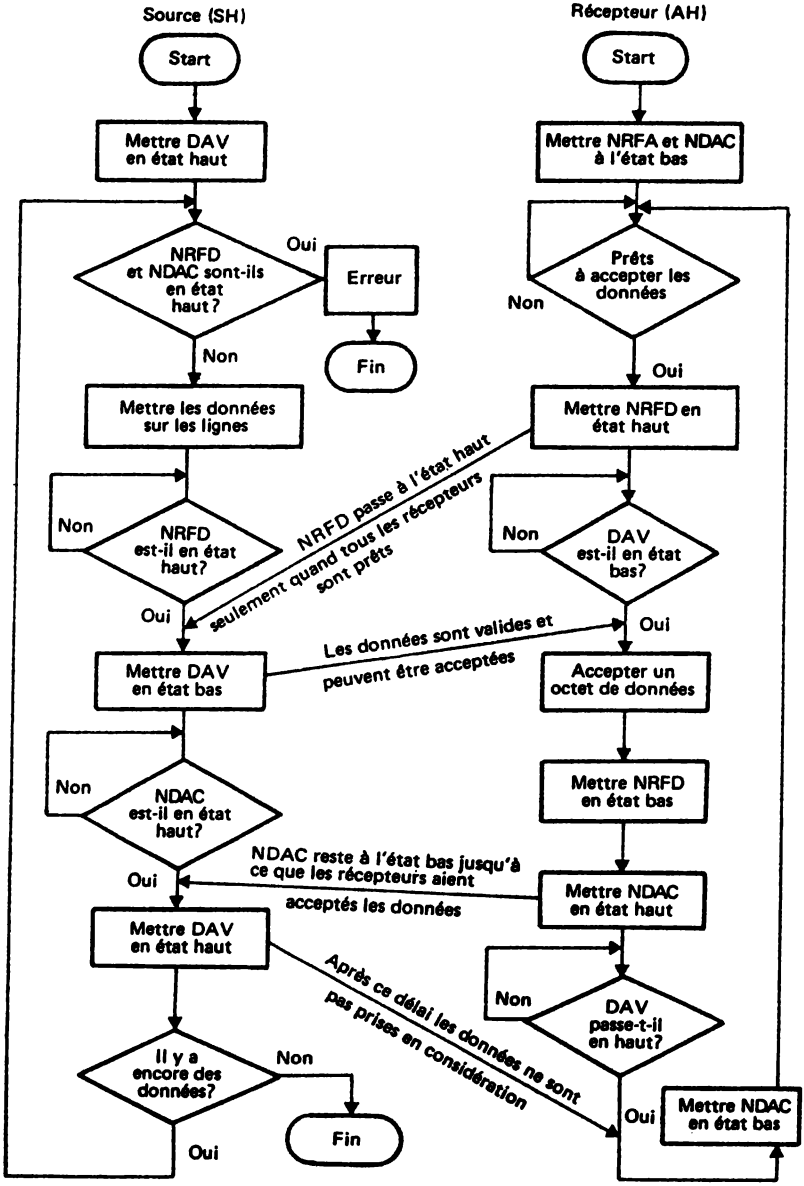


Fig. 6 — La procédure du dialogue.

4. ADRESSAGE ET DIALOGUE

L'adressage des appareils peut être assuré sur deux niveaux, *primaire* et *secondaire*. A l'adresse primaire correspond une « fonction simple » mais si celle-ci est « étendue », elle bénéficiera d'un second niveau d'adressage (secondaire). Le système comporte ainsi 31 adresses primaires et autant de secondaires. C'est au contrôleur actif qu'il revient d'adresser les appareils.

Pour cela, il émet un octèt comportant :

— un bit de parité (P) ;
— un code sur 2 bits définissant le mode ;
— 5 bits d'adresse.

Le code sur deux bits est le suivant :

— 01 indique qu'il s'agit d'une adresse primaire d'auditeur ;
— 10 indique que l'adresse primaire est celle d'un orateur ;
— 11 est formé pour une commande secondaire ; et
— 00 pour une commande auxiliaire.

Il est évident que si des appareils se voient attribuer une adresse commune. ils seront activés en même temps.

Dans le cas de commandes auxiliaires, un format sur un octet tel que P001 XXXX (avec P = parité) est une commande auxiliaire adressée.

La norme définit encore une liste de plusieurs dizaines de messages avec leur mnémonique.

Le cycle général de la procédure de dialogue est donné dans la figure 6 ; elle s'appuie sur les trois signaux DAV, NRDF et NDAC, le premier dépendant de l'orateur et les seconds des auditeurs. La figure 7 est le chronogramme type d'un échange en poignée de mains.

En fait, la description complète des commandes, de l'organisation du dialogue, des fonctions et des possibilités de ce bus requièrent une étude complète à laquelle les lecteurs intéressés devront se livrer. Ajoutons que divers équipements complémentaires ont été conçus pour le servir. Ainsi, par exemple, *Hewlett-Packard* a proposé, fin 1979, un système de liaison par fibres optiques permettant d'éloigner les appareils d'une centaine de mètres (interface HP 12050A).

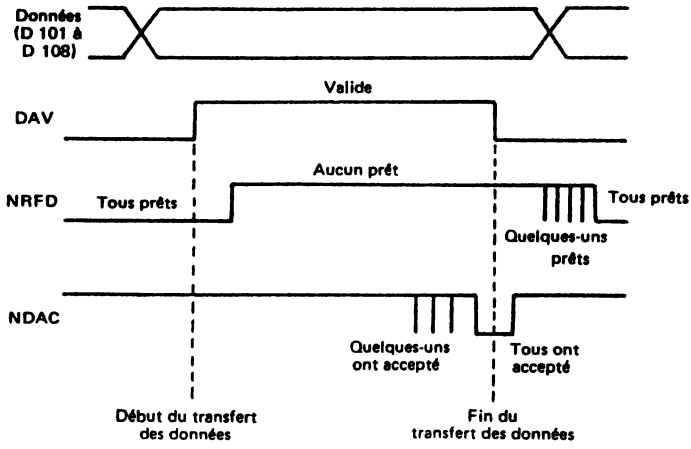


Fig. 7 — L'échange en poignée de mains.

5. CIRCUITS INTÉGRÉS SPÉCIALISÉS IEEE-488

Pour faciliter l'usage de ce bus normalisé, des interfaces intégrées ont été proposées.

Intel a ainsi réalisé le circuit 8291 à 40 broches, dont le synoptique est donné figure 8.

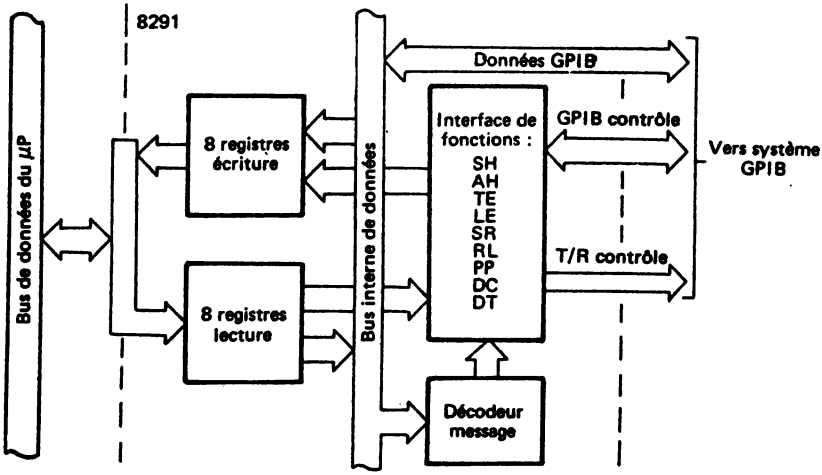


Fig. 8 — Synoptique du 8291 de Intel.

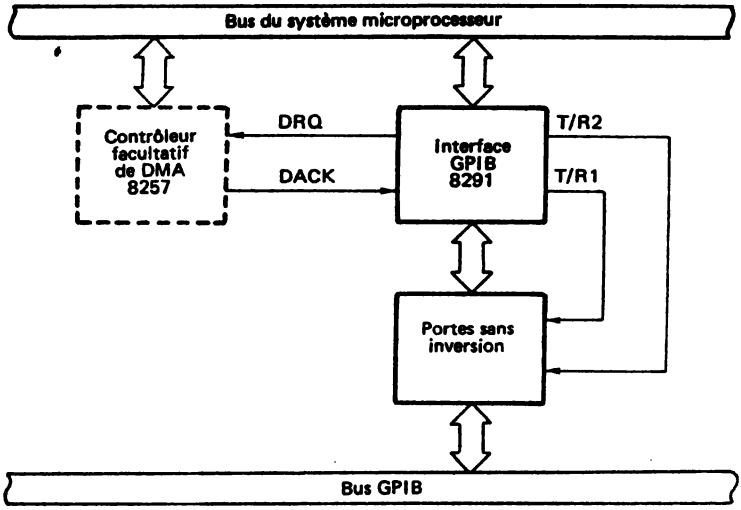


Fig. 9 — Intervention du 8291 dans un système.

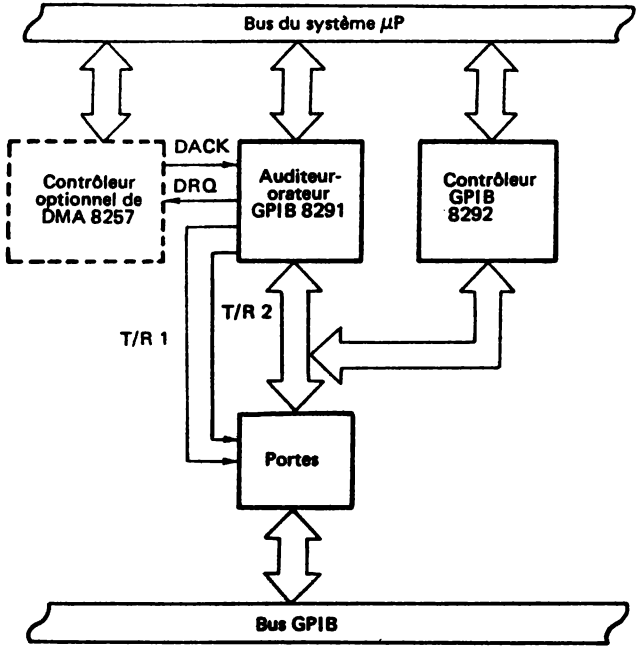
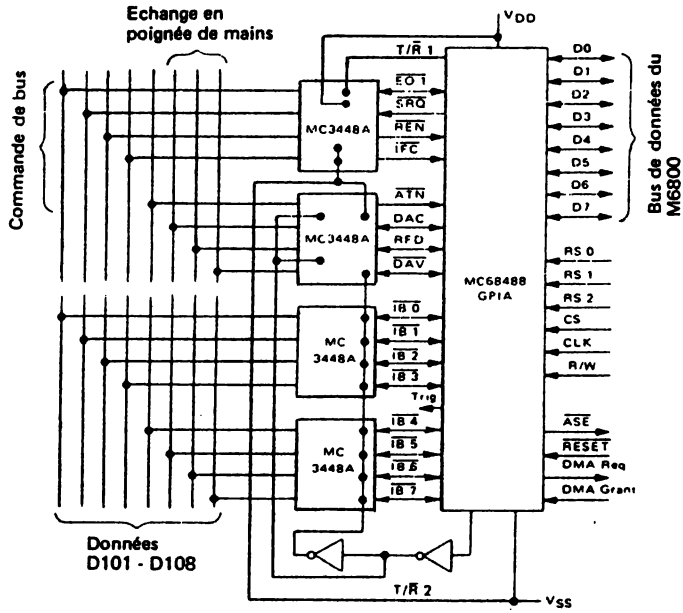


Fig. 10 — Association d'un 8292 et d'un 8291 : toutes les fonctions normalisées sont assurées.

Ce circuit communique avec les bus de microprocesseurs tels que les 8080, 8085, 8086, 8048... et exécute toutes les fonctions standards, à l'exception de celles du contrôleur. Il interviendra ainsi, comme le montre la figure 9, dans un système.

CIRCUIT 8291 ET 8292

Associé à ce 8291, le 8292 d'*Intel* exécute, lui, toutes les fonctions GP-IB (IEEE-488). Ce circuit est, en fait et lui-même, un microprocesseur de la famille des « UPI-41A » qui ont été spécialement conçus pour servir d'interface, et il a été programmé en conséquence. Un système regroupant ces deux circuits est présenté figure 9.



Structure-type du MC3448A

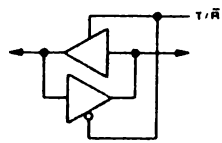


Fig. 11 — Intervention du MC 68488 de Motorola dans un système.

Le MC 6848 de *Motorola*, encore appelé GP-IA, se présente en un circuit à 40 broches qui exécute toutes les fonctions GP-IB, via des drivers de bus type MC 3448A, selon le synoptique de la figure 10. Il est connecté au bus de données d'un système à 6800, ou autre microprocesseur de la famille.

Texas Instruments a réalisé le TMS 9914, appelé GPIBA et répondant à l'IEE 488-1978. Il s'associe aux transmetteurs SN 75160/61 et exécute toutes les fonctions normalisées. Le HEF 4738 de *RTC/Signetics* est une autre interface GP-IB, mais la liste en est bien plus longue.

6. LE BUS HP-IL

HP-IL

L'intérêt du bus IEEE-488, proposé tout d'abord par *Hewlett-Packard*, est évident. Cependant, il n'est pas sans défaut : prix élevé, tout d'abord, puis complexité relative d'emploi. Aussi, cette même société a-t-elle annoncé, en 1981, un nouveau bus plus simple et plus économique, le HP-IL, pour « *Hewlett-Packard Interface Loop* », boucle d'interface Hewlett-Packard.

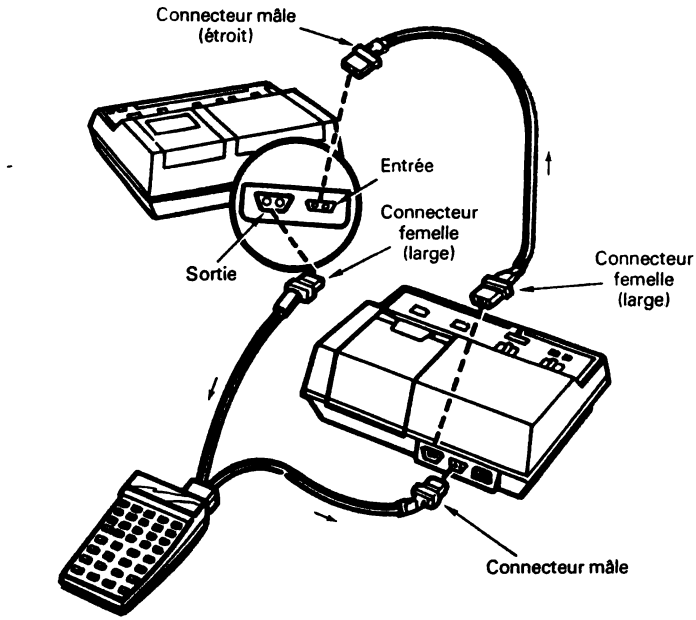


Fig. 12 — Exemple de liaison HP-IL, pilotée par un calculateur HP-41C. La liaison se fait en boucle.

Ce HP-IL est, lui, un bus de communication *série*, destiné, à l'origine, aux appareils portables fonctionnant sur batteries (fig. 12). Une des particularités de ce bus d'interface est l'utilisation d'une simple liaison deux fils torsadés ou blindés, pour l'acheminement des données, sur une centaine de mètres (en blindé), une dizaine (torsadé). La sortie de chaque appareil est reliée à l'entrée de l'appareil suivant pour former, d'appareil en appareil, une boucle fermée. Jusqu'à 31 dispositifs compatibles HP-IL peuvent ainsi être interconnectés.

Il y a trois types de dispositifs sur une boucle d'interface classique : un *contrôleur*, des *émetteurs* (« talkers ») et des *récepteurs* (« listeners »). Le contrôleur, comme par exemple le calculateur HP 41C, gère l'échange de données entre les différents dispositifs connectés sur la boucle et fournit toutes les commandes de configuration d'interface. Un dispositif qui envoie des données est dit « émetteur », et un dispositif destiné à recevoir les données (par exemple une imprimante) est dit « récepteur ». Un instrument tel que le multimètre numérique HP 3468A est à la fois « émetteur » et « récepteur ». Les commandes envoyées sur la boucle par le contrôleur n'autorisent qu'un émetteur actif à la fois, mais il peut cependant y avoir plusieurs récepteurs simultanément.

Afin d'améliorer la « convivialité » du système, l'adresse de tout dispositif HP-IL peut être assignée de façon automatique, et chaque dispositif est capable d'informer le contrôleur de sa fonction propre. Cette fonction d'adressage automatique du HP-IL permet à l'utilisateur d'adresser les périphériques soit par leur code, soit par leur nom générique. Tous les dispositifs de la boucle reçoivent leur adresse du contrôleur à chaque mise sous tension ou à la suite d'une commande logicielle.

Un seul message à la fois est transmis sur la boucle, celui-ci devant revenir au dispositif qui l'a émis avant qu'un autre message puisse être transmis. Cette caractéristique assure que *la vitesse de transmission sur la boucle ne sera jamais supérieure à la vitesse acceptable par le dispositif le plus lent du système.* Tous les messages doivent transiter par chaque dispositif, et un dispositif donné ne transmettra le dernier message reçu que lorsqu'il sera prêt à recevoir le message suivant. Une autre caractéristique intéressante de cette configuration est le fait que le dispositif étant à l'origine d'un message peut, avant d'envoyer le message suivant, vérifier la concordance entre le message émis et le message en retour, et s'assurer ainsi que tous les dispositifs sur la boucle ont reçu et retransmis ce message sans erreur.

6.1. Transmission des messages

Tout ce qui précède suggère que le HP-IL n'est pas très rapide. Naturellement, si un certain nombre de dispositifs lents sont présents sur la boucle, la vitesse de transmission peut devenir inacceptable. Cependant, un message n'est normalement destiné qu'à un ou deux dispositifs à la fois, et l'interface HP-IL est conçue de telle façon qu'un message qui n'est pas destiné à un dis-

positif donné est retransmis par celui-ci dans les 20 µs suivant sa réception. Cette caractéristique autorise une vitesse de transmission acceptable.

Lorsqu'une commande du contrôleur est destinée à tous les dispositifs de la boucle, la procédure de contrôle de séquence est modifiée, et chaque dispositif recevant cette commande la stocke dans une mémoire tampon et la transmet au dispositif suivant avant même de l'exécuter. Ainsi, tous les dispositifs sur la boucle exécuteront la commande pratiquement au même instant. Le contrôleur envoie ensuite un message spécial appelé « prêt pour ordre » qui est retenu par chaque dispositif tant que la commande précédente n'a pas été exécutée. Tous les dispositifs demandant approximativement le même temps pour exécuter une commande donnée, « prêt pour ordre » est retardé seulement par le premier dispositif sur la boucle, pour être ensuite transmis rapidement d'un dispositif à l'autre.

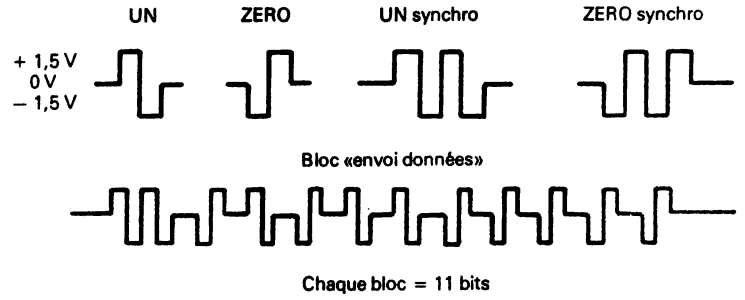


Fig. 13 — Codage en HP-IL. Chaque message est transmis sous forme d'un bloc de 11 bits. Tous les bits sont codés sur trois niveaux (codage bipolaire) et la largeur d'une impulsion est de 1 µs. Un bloc complet est transmis en 46 µs, et la transmission des blocs est asynchrone.

Chaque message est transmis sur la boucle sous forme d'un bloc de onze bits, le premier bit étant un bit de synchronisation permettant de reconnaître le début du bloc. Le bit de synchronisation et les deux bits qui le suivent indiquent le type de message, par exemple une commande concernant les données. Les huit bits suivants constituent le message proprement dit. Tous les bits sont codés sèlon trois niveaux à – 1,5 V, 0 V et + 1,5 V (codage bi-polaire) comme indiqué sur la figure 13, chaque impulsion ayant une largeur nominale d'une microseconde. Un bloc complet est transmis en 46 µs et la transmission des blocs est asynchrone. La vitesse maximale théorique de transmission de la boucle est légèrement supérieure à 20 K/s mais si l'on tient compte des retards normaux occasionnés aussi bien par le logiciel que par le matériel, une vitesse de l'ordre de 3 à 5 Ko/s est certainement plus réaliste. Un multimètre tel que le 3468A, par exemple, envoie 13 blocs pour chaque lecture, signe et exposant compris.

6.2. Exemple d'application avec un multimètre

Supposons qu'un utilisateur désire constituer un système simple comprenant un calculateur HP-41C, une imprimante thermique HP 82162A et un multimètre numérique HP 3468A, le multimètre effectuant des mesures dont le résultat doit être imprimé. Ce système est illustré sur la figure 14. Le calculateur configure automatiquement la boucle, et le premier dispositif (l'imprimante) reçoit l'adresse « 1 », et le deuxième dispositif (le multimètre) reçoit l'adresse « 2 ». Le multimètre ayant été sélectionné, la mesure effectuée doit être placée dans le registre X (numérique) du calculateur. La commande IND (entrée décimale) est utilisée dans le programme, celle-ci provoquant l'envoi sur le bus de la commande « envoi données ». Lorsque cette commande est reçue par le multimètre, elle n'est pas retransmise, et celui-ci commence l'envoi du résultat de la mesure sous forme d'une suite de blocs de données terminée par un « retour chariot/saut de ligne ». Pendant le processus de transmission des données, le 41C place la mesure dans son registre X.

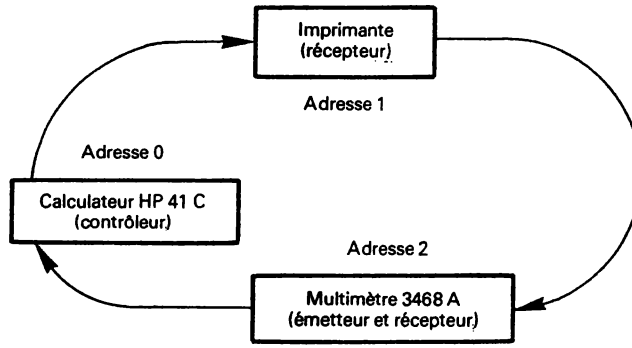


Fig. 14 — Exemple de chaîne incorporant un HP-41C, une imprimante et un multimètre.

La structure de la boucle permet au multimètre d'effectuer un contrôle d'erreur avant l'envoi du bloc de données suivant. Si une erreur est détectée, la séquence est terminée par un « fin de transmission-erreur ». Lorsque ce message est reçu par le 41C, le programme est interrompu et l'affichage du calculateur indique « TRANSMIT ERR ».

La commande PRX est utilisée pour l'impression du contenu du registre X. Une reconnaissance série est effectuée sur la boucle jusqu'à ce que l'imprimante soit reconnue. Le calculateur donne alors l'ordre à l'imprimante de passer en mode *récepteur*, et la séquence stockée dans le registre X est envoyée à l'imprimante. La vitesse de ce système est limitée par le HP-41C à environ 2 lecteurs par seconde.

CHAPITRE XII

INTERFAÇAGE AVEC LE MONDE ANALOGIQUE

A notre échelle, les phénomènes physiques nous apparaissent comme des grandeurs analogiques, c'est-à-dire à évolution continue (et non numérique) : température, pression, accélération, etc. Pour les traiter avec un système à microprocesseur, travaillant par conséquent en numérique, il faudra assurer une conversion analogique-numérique à l'entrée, numérique-analogique à la sortie si les besoins le requièrent.

Les mots-clés de ce chapitre

A/N : conversion analogique-numérique, 216
A/D : conversion analogique-digitale, 216
CAN : convertisseur analogique-numérique, 216
CAD : convertisseur analogique-digital, 216
N/A : conversion numérique-analogique, 216

D/A : conversion digital-analogique, 217
CNA : convertisseur numérique-analogique, 217
CDA : convertisseur digital-analogique, 217
ADC : convertisseur analogique-numérique, 217
DAC : convertisseur digital-analogique, 217

Rampe : mode de conversion A/N, 221
Comptage : mode de conversion A/N, 225
Approximations successives : mode de conversion A/N, 227
Flash : mode de conversion A/N, 230
Parallèle : mode de conversion A/N, 230

Résistances pondérées : mode de conversion N/A, 232
Echelle R/2R : mode de conversion N/A, 234
Circuit ADC 80 : convertisseur intégré A/N, 237
Monostable : mode conversion A/N, 238
Manettes de jeux pour l'IBM-PC, 238

1. INTRODUCTION

Les grandeurs physiques qui nous intéressent quotidiennement sont, à notre échelle, analogiques. Pour les prélever, on sert de *capteurs* ; le rôle du capteur consiste à traduire une grandeur physique quelconque (pression, température, débit...) par un signal électrique exploitable. Actuellement, et probablement encore pour longtemps, les capteurs sont en majorité du type analogique également, ce qui signifie qu'ils fournissent un signal électrique analogue à la grandeur mesurée.

Ce signal doit, ensuite, être exploité et, pour cela, on passe par un système de traitement numérique, ce qui impose une première conversion analogique à numérique. Après traitement, le système numérique, selon les cas, commande des dispositifs d'arrêt-marche, des affichages, des régulations, des enregistrements, etc. S'il s'agit d'une commande de régulation, on pourra être amené à agir sur des *actionneurs* parfois, eux aussi, analogiques (par exemple, une vanne analogique). Par conséquent, le signal numérique issu du système de traitement devra être reconverti en analogique.

Le même problème se posera si l'on souhaite actionner un indicateur à aiguille. Ou encore, et en téléphonie, lorsqu'on passera par une transmission numérique de la parole, avec un microphone comme capteur et un écouteur en fin de chaîne.

Les télécommunications numériques, l'enregistrement numérique du son et des images, constituent en effet des domaines extrêmement vastes d'application des convertisseurs, et il se prépare là une véritable révolution. Le plus souvent, ces convertisseurs œuvreront également en compandeurs (« compresseurs-expanseurs ») selon des lois logarithmiques et seront incorporés à des systèmes appelés CODEC, pour « codeur-décodeur ». Le codage est assuré selon la technique dite de modulation d'impulsions (MIC).

A/N
A/D
CAN
CAD
N/A

De ce fait, une chaîne-type incorporant tous ces éléments peut être conçue comme le montre la figure 1. Les informations disponibles à la sortie du système de traitement apparaissent en numérique et peuvent être utilisées directement par des dispositifs numériques, ou après une ultime conversion numérique-analogique pour les dispositifs analogiques.

On notera, à ce propos, qu'on utilise couramment une série d'abréviations qui sont :

- A/N, pour conversion analogique-numérique ;
- A/D, synonyme de A/N, pour analogique-digital ;

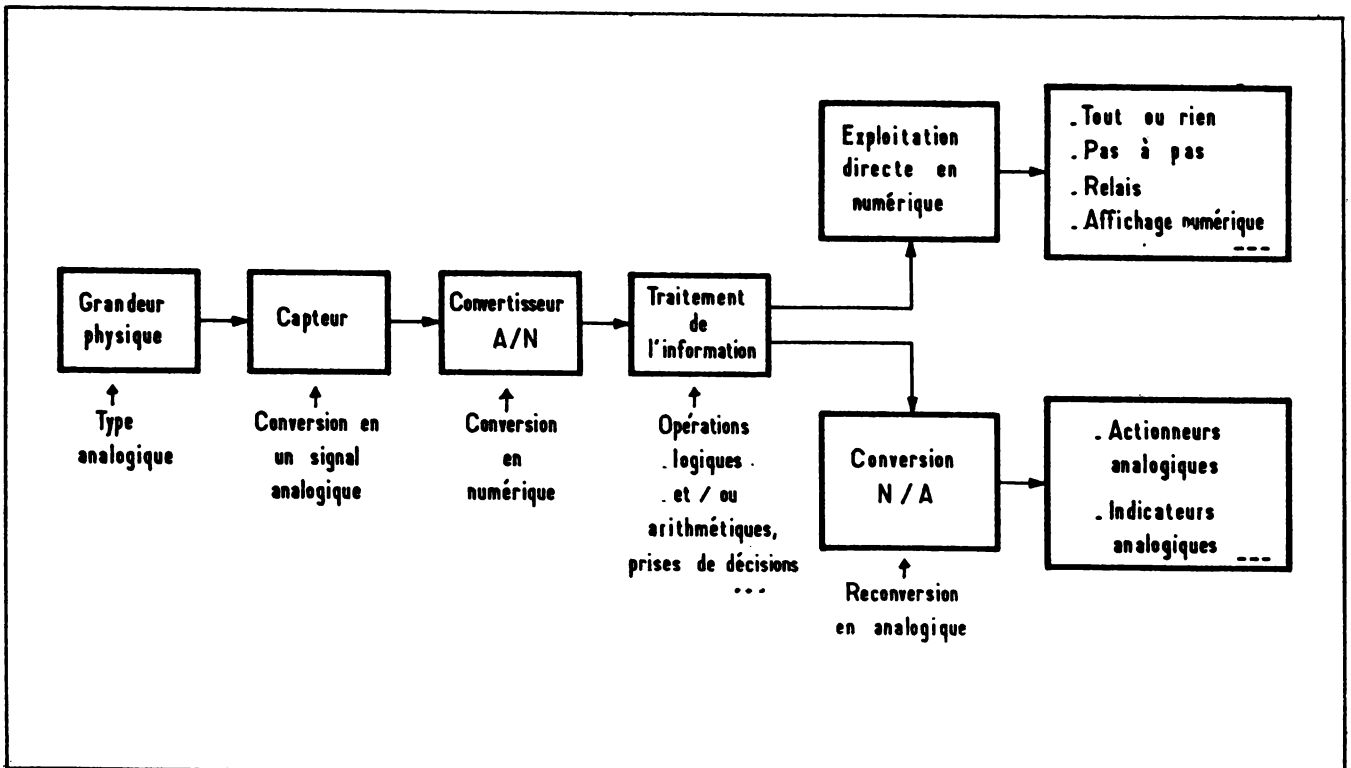


Fig. 1. — Organisation type d'une chaîne industrielle faisant intervenir des convertisseurs.

D/A
CNA
CDA
ADC
DAC

• CAN ou CAD, pour convertisseur analogique-numérique (ou digital) ;
• N/A ou D/A, pour conversion numérique-analogique ;
• CNA ou CDA, pour convertisseur numérique-analogique (ou digital) ;
• ADC ou DAC, si l'on se réfère à la terminologie anglo-saxonne, pour *analog to digital converter* et *digital to analog converter.*

2. CONVERSION ANALOGIQUE-NUMÉRIQUE

CONVERSION A/N

Commençons par la première partie de ce synoptique, relative à la conversion A/D. On peut développer cette section afin de cerner de plus près sa structure, ce qui mène au schéma de la figure 2. Le paramètre physique est saisi par un transducteur, ou capteur, qui fournit un signal analogique estimé trop faible et donc amplifié. Parce que plusieurs grandeurs physiques peuvent témoigner de l'état d'un système à un moment donné (température, niveau et viscosité, par exemple), on a imaginé ici qu'on disposait de multiples informations analogiques, prélevées éventuellement selon une chaîne identique capteur-amplificateur ; si le système de traitement de l'information ne peut en accepter qu'une à la fois, on passe par un *multiplexeur* : c'est un dispositif qui reçoit toutes les entrées possibles mais ne livre la voie unique de sortie qu'à l'une d'elles, sous la commande de circuits non représentés. La vitesse de scrutation (de balayage) des voies dépend surtout de la technologie du multiplexeur.

Ce dernier est suivi par un circuit qui prélève, sur chaque voie, un échantillon du signal et le met en mémoire provisoirement ; c'est le circuit d'échantillonnage et mémorisation. Au moment opportun, cet échantillon est transmis via un buffer au convertisseur A/N qui exécute la conversion, toujours sous commande de circuits non représentés ici ; on peut imaginer, par exemple, que ces commandes proviennent d'un microprocesseur.

On ajoutera encore que :

• l'amplificateur d'entrée améliore l'immunité aux bruits, porte le signal utile au bon niveau et adapte les indépendances ;

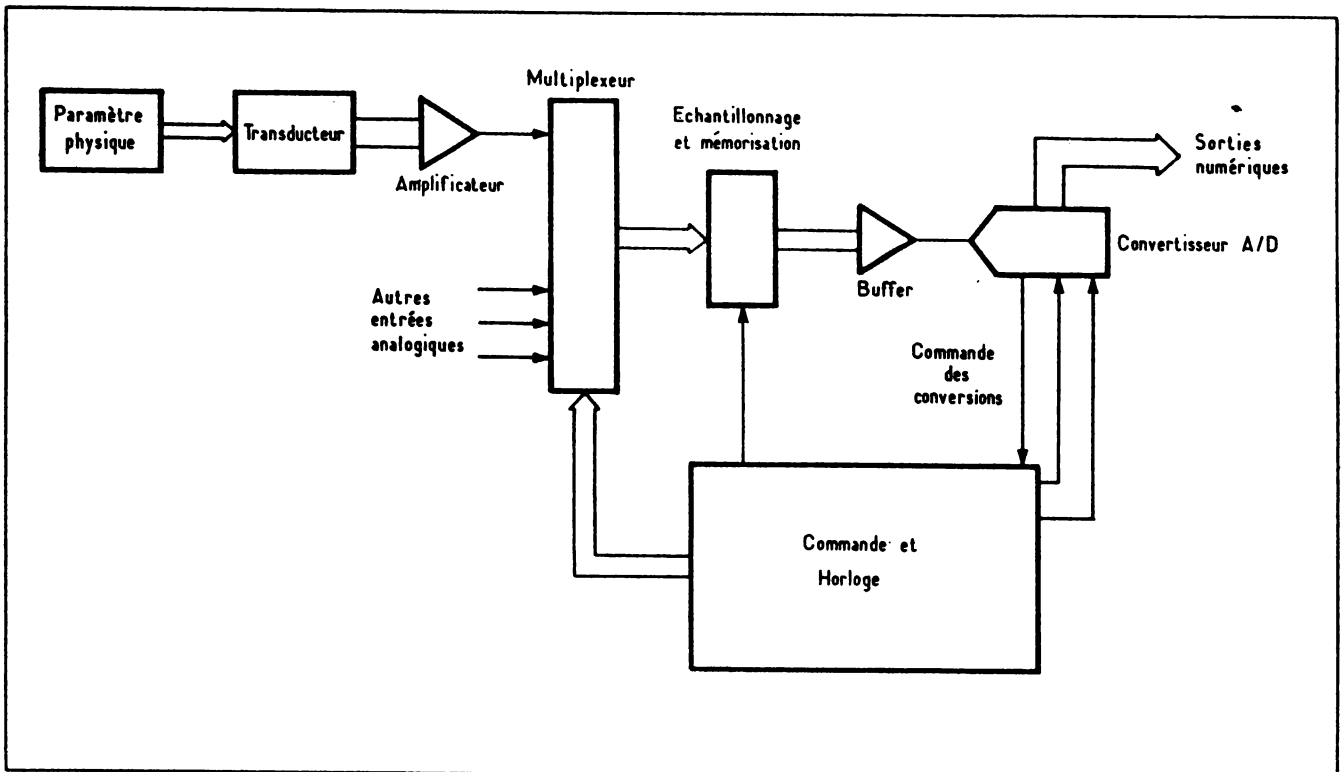


Fig. 2. — Circuits intervenant dans une chaîne type d'acquisition d'informations analogiques.

Exemple de quelques capteurs types

Grandeurs	Capteurs
Température	Thermocouples Thermistances Semiconducteurs
Pression	Potentiométriques Inductifs Capacitif Piezorésistif Jauges résistantes
Position	Potentiométriques Inductifs Optiques Effet Hall Magnétodiode
Débit	Pression différentielle Ultrasons Turbine (comptage) Transfert thermique

• le multiplexeur sélectionne une ligne parmi N lignes ;

• l'échantillonneur fournit une valeur moyenne pendant le temps de la mise en service au convertisseur ;

• le buffer qui précède ce dernier sert de tampon ; il réduit les fuites capacité-mémoire qui stocke temporairement l'échantillon et possède une grande impédance d'entrée, mais attaque le convertisseur à basse impédance.

L'échantillonnage et la mémorisation du signal à convertir ne sont utiles que pour autant que celui-ci soit à évolution rapide. Si l'évolution du signal est lente, ce qui est généralement le cas des températures, ces circuits sont inutiles, l'erreur introduite restant souvent négligeable.

Pour procéder à la conversion en numérique, plusieurs méthodes peuvent être adoptées ; elles offrent toutes des avantages et des inconvénients. Les quatre principales d'entre elles sont :

1) *à rampe*, très « classique » ;
2) *les approximations successives*, plus rapide que la précédente ;
3) *le comptage*, très économique ;
4) *flash* ou *parallèle* : la plus rapide et la plus chère.

De nombreuses variantes de ces méthodes existent, en outre : par exemple, la *simple rampe*, la *double rampe* ou la *triple rampe*, pour la première citée...

Dans le cas de conversion sans besoin de précision ni vitesse, on pourra exploiter des principes plus simples et plus économiques. Ainsi, la résistance des potentiomètres de manettes de jeux peut fort bien piloter la constante de temps d'un monostable, qui enverra ensuite une impulsion (cas appliqué sur l'IBM-PC).

2.1. Conversion à rampe

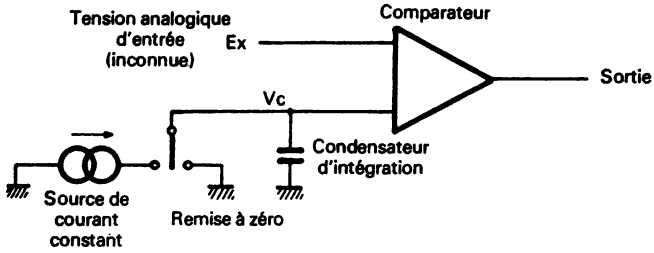
CONVERSION A RAMPE

Le convertisseur à rampe unique est l'un des plus simples qui soient, mais il souffre d'un manque de précision. Le principe en est le suivant : on charge linéairement un condensateur aux bornes duquel on obtient une rampe de tension (fig. 3) ; cette tension est appliquée à l'entrée d'un comparateur dont l'autre entrée reçoit la tension inconnue E_x ; lorsque la rampe arrive à égalité avec E_x, la sortie du comparateur bascule. Le temps écoulé entre le départ de la rampe et le basculement est mesuré par un compteur qui fournit ainsi, sous forme numérique, l'équivalent de la grandeur analogique d'entrée. Pour cela, on se servira évidemment d'une horloge de référence. Avant le départ de la rampe, on remet le niveau de charge du condensateur à zéro, comme le montre la figure, grâce à un commutateur à deux positions.

Parce que ce convertisseur se ramène à une mesure du temps, on l'appelle encore *convertisseur tension-temps*, ou encore *convertisseur à modulation de largeur d'impulsion* eu égard à son principe de fonctionnement (fig. 4).

Il est astucieux de faire correspondre le nombre d'impulsions d'horloge comptées avec la gamme de mesure. Si, par exemple, la gamme des tensions d'entrée peut varier de 0 à 10 V, on s'arrangera pour que la fréquence d'horloge et la vitesse de montée de la rampe fassent correspondre une impulsion à un millivolt, par exemple. Ainsi, les 10 V se traduisent par le comptage de 10 000 impulsions ; on peut alors procéder à un affichage direct, si besoin est. Si l'horloge est à 1 MHz, il faudra 0,01 s pour compter ces 10 000 impulsions, soit 10 ms : ce temps est relativement long ; on en conclut que la conversion à intégration est lente.

D'autre part, la conversion est peu précise, pratiquement de l'ordre de 10^{-3}. Aussi, et bien que ce convertisseur soit simple et économique, lui préfère-t-on le « double rampe ».



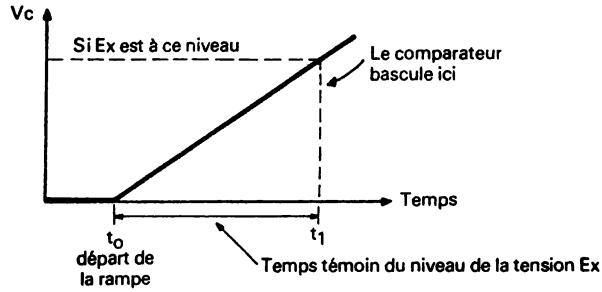


Fig. 3 — Principe de la conversion à intégration simple rampe. Le temps nécessaire pour atteindre l'égalité avec la tension analogique d'entrée à convertir témoigne de cette tension.

Le convertisseur double rampe constitue un perfectionnement important du simple rampe : la précision devient indépendante de la capacité et de la fréquence d'horloge. La conversion s'effectue ici en deux temps :

1) La tension inconnue E_x est appliquée à l'entrée d'un intégrateur pendant un temps prédéterminé, mesuré par un compteur.

2) L'entrée de l'intégrateur est ensuite commutée sur une tension de référence E_{ref} dont la polarité est opposée à celle de E_x. Sa tension de sortie décroît alors linéairement jusqu'à zéro. Le temps de décroissance donne la mesure de la tension inconnue E_x.

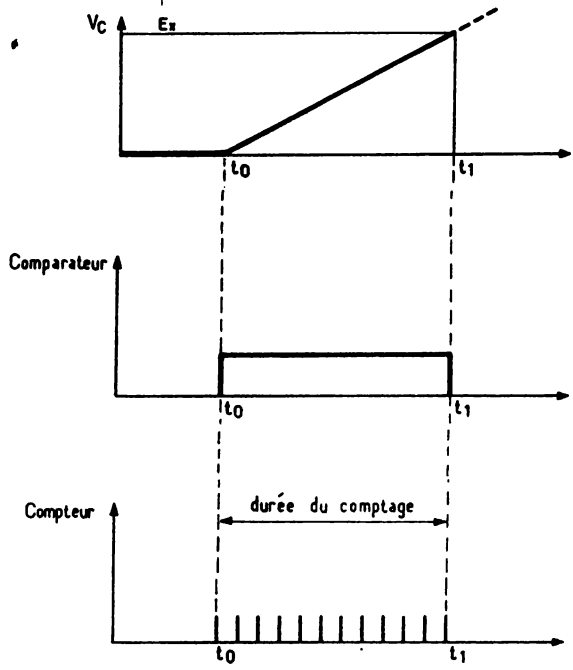


Fig. 4 — Avec le simple rampe, le temps écoulé jusqu'à ce que le comparateur bascule constitue l'équivalent de la grandeur analogique d'entrée.

Le synoptique d'un tel convertisseur est donné figure 5 avec sa double rampe. La mesure de la tension inconnue est fournie par l'équation :

$$E_x = E_{ref}\,(N_1/N_2),$$

où N_1 est le nombre d'impulsions compté pendant l'ascension de la rampe, et N_2 celui compté sur l'autre versant. On constate que le condensateur n'intervient plus ; la double intégration compense l'erreur qu'il pourrait introduire.

De plus, l'intégration de la tension inconnue réduit les erreurs dues aux bruits. Bien mieux, si le temps imparti à la première intégration est de 20 ms, soit la durée d'une période à 50 Hz, ou encore un multiple de cette valeur, on obtient une réjection parfaite des signaux alternatifs à cette fréquence (en effet, l'intégration d'un signal sinusoïdal pendant une période donne un résultat nul), ou à ses multiples.

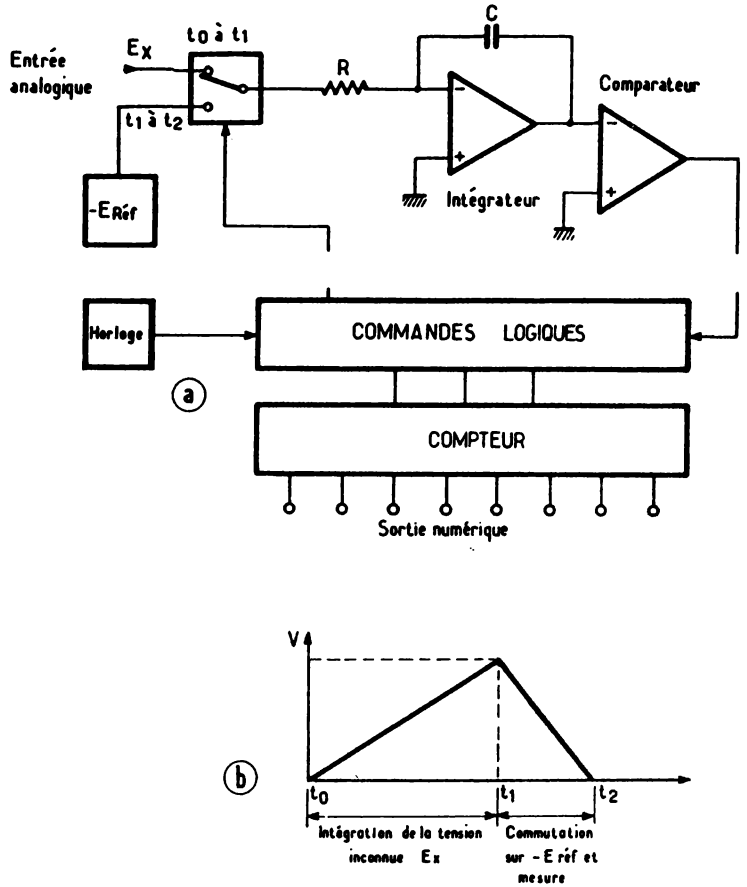


Fig. 5 — Organisation d'un convertisseur à intégration double rampe (a), et aspect des rampes (b).

La linéarité différentielle des deux intégrations est, elle, excellente. Au total, on obtient ainsi une grande précision, de l'ordre de 10^{-4} à 10^{-5} de la pleine échelle dans les meilleurs cas. En revanche, la vitesse de conversion est encore plus faible qu'avec le simple rampe : si l'on veut conserver la réjection du 50 Hz, on exécutera typiquement de 5 à 30 conversions par seconde.

Si l'on se réfère à la figure 6, où les courbes d'intégration ont été dessinées pour deux valeurs de la tension E_x, on constatera encore que le temps d'intégration de E_x reste constant, quelle que soit sa valeur, alors que la seconde intégration s'effectue, elle, à pente constante.

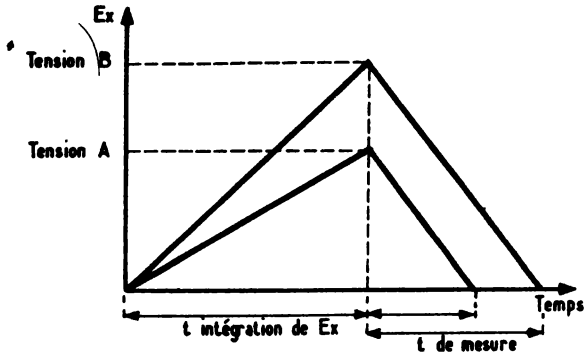


Fig. 6 — Dans le double rampe, la première est à temps constant et la seconde à pente constante. On appelle encore ce convertisseur « double pente ».

En résumé, ce convertisseur est simple et précis, économique mais lent, est unipolaire dans sa version de principe (c'est-à-dire qu'il n'accepte de signaux d'entrée que d'une seule polarité) ; sa sortie est *monotone* par construction (le mot de sortie n'est jamais accidentellement inférieur, à un instant donné, pour une tension d'entrée croissante). De plus, il procure une excellente réjection des bruits, ses composants ne sont guère critiques, et il permet la réalisation aisée de circuits de remise à zéro automatique. Par contre, le temps de conversion est variable et dépend de la tension d'entrée.

2.2. Convertisseur à comptage

CONVERSION A COMPTAGE

Le convertisseur par comptage est encore appelé *convertisseur incrémental.* A la différence des schémas précédents, on ne procède plus par une conversion analogique-numérique, mais bien à une conversion numérique-analogique dont le résultat est comparé à la tension inconnue d'entrée.

Le synoptique d'un convertisseur par comptage est donné figure 7. Au départ de la conversion, on applique des impulsions d'horloge à un compteur binaire ; la sortie de ce compteur binaire attaque un convertisseur numérique-analogique qui fournit donc une tension analogique correspondante à la sortie : cette tension croît en marche d'escalier, comme le montre la courbe, bit après bit. Elle est comparée à la tension inconnue par un amplificateur opérationnel qui assure la sommation des deux tensions et agit par conséquent en

comparateur, ou encore directement par un comparateur attaqué sur chacune de ses entrées par les deux tensions. A l'égalité, l'horloge est stoppée et la valeur fournie par le compteur binaire est transmise aux sorties.

La précision est, ici, indépendante du temps et liée uniquement à celle du convertisseur numérique-analogique ; elle est généralement très bonne, et d'autant meilleure que le comptage porte sur une valeur plus élevée (incréments plus petits). Malheureusement, sa durée augmente alors, ce qui fait de ce convertisseur un système lent. On peut ajouter qu'il est simple et économique.

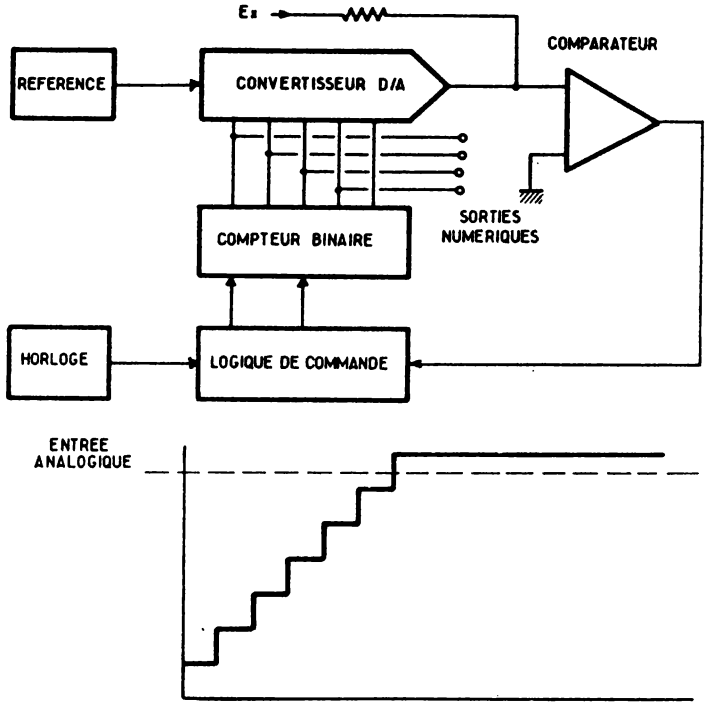


Fig. 7 — Principe du convertisseur incrémental (à comptage) : chaque marche de l'escalier est convertie en son équivalent analogique, lequel est comparé à la tension analogique d'entrée.

En variante, il peut fonctionner avec un compteur réversible, comptant ou décomptant à partir d'une valeur initiale, ou même avec deux compteurs associés dont l'un prend le relais de l'autre : si les incréments du second sont plus faibles, on pourra obtenir une mesure rapide à l'aide du premier, plus lente mais plus fine à l'aide du second.

Ce type de convertisseur, enfin, peut intervenir dans une chaîne complète en agissant comme convertisseur A/N à l'entrée, et pour sa section CNA uniquement, comme convertisseur N/A à la sortie, ce qui se révèle astucieux et économique.

2.3. Convertisseur à approximations successives

CONVERSION A APPROXIMATIONS SUCCESSIVES

Avec le convertisseur à approximations successives, on compare la tension inconnue à des tensions successives de référence, un peu comme l'on pèse une marchandise quelconque sur une balance en jetant des poids dans l'un des plateaux. C'est d'ailleurs pourquoi on l'appelle également *convertisseur à pesées.*

Pour bien en comprendre le fonctionnement, on va l'illustrer avec un exemple : soit à mesurer une tension d'entrée de 7 V avec un convertisseur couvrant l'échelle de 0 à 10 V :

1) le premier « poids » appliqué au comparateur (qui reçoit les 7 V sur sa seconde entrée) est la moitié de l'échelle, soit 5 V ; ce poids est *inférieur* à +7 V et est *retenu* ;

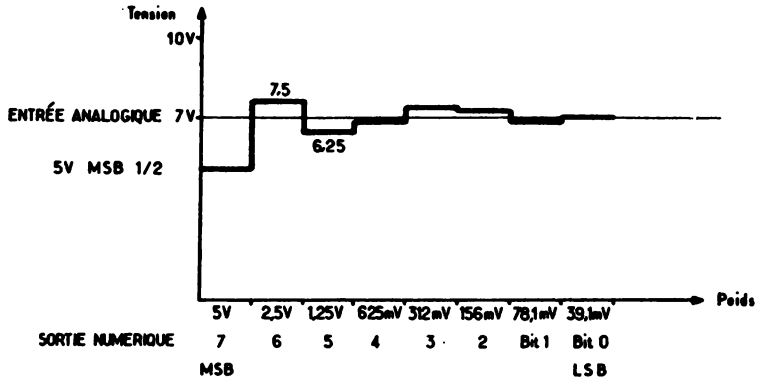
2) la valeur moitié de ce premier poids, +5 V, est de 2,5 V. On applique 2,5+5 V à l'entrée du comparateur qui reçoit toujours la tension à convertir, +7 V, sur l'autre entrée. Parce que +7,5 V est *supérieur* à 5 V, on *rejette* la seconde tension d'essai de 2,5 V ;

3) la valeur moitié de 2,5 V est 1,25 V. On applique alors 1,25 V+5 V (retenus précédemment) au comparateur. Cette fois, ces 6,25 V sont *inférieurs* à 7 V ; on *retient* donc 1,25 V ;

4) la valeur moitié de 1,25 V est 625 mV ; on applique 0,625 V+1,25 V+5 V=6,875 V au comparateur. Cette valeur étant *inférieure* à 7 V, on *retient* encore 0,625 V ;

5) ainsi de suite jusqu'à la précision requise.

La somme des tensions retenues, en numérique, est la conversion de la tension analogique d'entrée. La courbe de la figure 8 illustre ce processus, alors que la figure 9 symbolise la structure de ce convertisseur dans lequel intervient, en fait, un convertisseur numérique-analogique. A chaque poids retenu correspond un 1 binaire ; tous les bits, 0 ou 1, sont regroupés en un mot, d'un octet ici, dans un registre.



COMPARAISONS POUR 7V		
1	5	CONSERVÉ
2	+ 2.5 = 7.5	ELIMINÉ
3	5.0 + 1.25 = 6.25	CONSERVÉ
4	+ 0.625 = 6.875	CONSERVÉ
5	+ 0.312 = 7.187	ELIMINÉ
6	6.875 + 0.156 = 7.037	ELIMINÉ
7	6.875 + 0.078 = 6.953	CONSERVÉ
8	+ 0.039 = 6.992	CONSERVÉ RESULTAT

Fig. 8 — Principe du convertisseur à approximations successives : on jette des « poids » dans une balance et on en fait la somme.

Très répandu en raison de sa vitesse de conversion (elle est fixe pour chaque bit et peut descendre à 100 ns par bit), ce convertisseur est aussi économique. Sa précision dépend surtout de la stabilité de la tension de référence, de la qualité de son CNA, des caractéristiques de commutation, de la précision du comparateur. Dans l'exemple illustré par la figure, on a développé une séquence en huit comparaisons car, en effet, il est probable qu'une neuvième avec moins de 20 mV aura outrepassé les possibilités du comparateur, compte tenu de son offset.

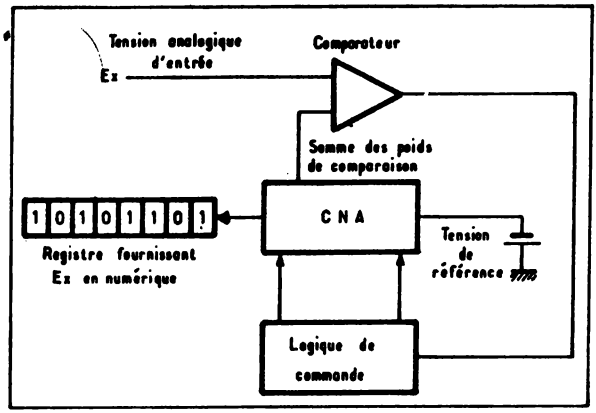


Fig. 9 — Structure type d'un convertisseur à approximations successives.

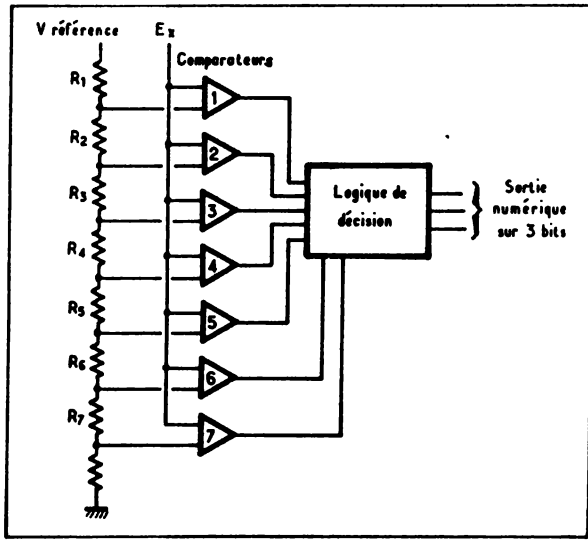


Fig. 10 — Le plus rapide, mais le plus complexe et le plus cher : le convertisseur parallèle. Les comparateurs en parallèle étudient l'entrée à partir de poids de référence ; une logique de sortie traite leurs résultats.

Typiquement, ce convertisseur fournira 50 000 lectures par seconde, mais il exige des composants critiques et est plus onéreux. D'autre part, il n'est pas monotone par construction comme le double rampe.

2.4. Convertisseur flash, ou parallèle

CONVERSION FLASH

CONVERSION PARALLELE

Le convertisseur à approximations successives est rapide, mais ce qui le freine c'est qu'il ne dispose que d'un seul comparateur. C'est lui qui exécute les comparaisons, l'une après l'autre, en série par conséquent.

Si l'on consent à multiplier le nombre de comparateurs, on pourra réaliser un convertisseur parallèle, bien plus rapide. Chaque convertisseur est polarisé à une tension de référence à partir d'une chaîne de division à résistances.

Le schéma d'un tel convertisseur est donné figure 10. La tension analogique d'entrée E_x est appliquée à tous les comparateurs, en parallèle ; certains de ceux-ci vont basculer ; leurs sorties sont combinées dans une logique de décision qui fournira, dans le cas de 7 comparateurs, un mot binaire sur 3 bits. La vitesse est telle qu'une conversion complète ne demande que quelques dizaines de nanosecondes. Par contre, le montage est plus important et plus onéreux, car le nombre de comparateurs (et de résistances...) croît très vite, et est de 2^{N-1} si N est le nombre de bits. Ainsi, une conversion sur 4 bits demande 15 comparateurs, alors que 5 bits en exigeraient 31, etc.

3. CONVERTISSEURS NUMERIQUE-ANALOGIQUE

CONVERSION N/A

De nombreux principes ont pu être adoptés pour assurer la conversion numérique-analogique. En pratique, on emploie essentiellement des convertisseurs du type parallèle (les bits du mot numérique d'entrée sont appliqués en parallèle) ; parmi ceux-ci, on retiendra les convertisseurs :

- à résistances pondérées ;
- à échelle R-2R.

Ces convertisseurs sont, par principe, plus rapides que ceux du type série mais exigent davantage de composants. La figure 11 symbolise la différence entre ces deux méthodes.

Résolution des CNA en fonction de la longueur du mot

Nombre de bits (N)	Résolution				
	en $1/2^N$	en décimal	en %	en ppm	en dB
1	1/2	0,5	50	500 000	− 6
2	1/4	0,25	25	250 000	− 12
3	1/8	0,125	12,5	125 000	− 18,1
4	1/16	0,0625	6,2	62 500	− 24,1
5	1/32	0,03125	3,1	31 250	− 30,1
6	1/64	0,015625	1,6	15 625	− 36,1
7	1/128	0,007812	0,8	7 812	− 42,1
8	1/256	0,003906	0,4	3 906	− 48,2
9	1/512	0,001953	0,2	1 953	− 54,2
10	1/1 024	0,0009766	0,1	977	− 60,2
11	1/2 048	0,00048828	0,05	488	− 66,2
12	1/4 096	0,00024414	0,024	244	− 72,2
13	1/8 192	0,00012207	0,012	122	− 78,3
14	1/16 384	0,000061035	0,006	61	− 84,3
15	1/32 768	0,0000305176	0,003	31	− 90,3
16	1/65 536	0,0000152588	0,0015	15	− 96,3
17	1/131 072	0,00000762939	0,00008	7,6	− 102,3
18	1/262 144	0,000003814697	0,0004	3,8	− 108,4
19	1/524 288	0,000001907349	0,0002	1,9	− 114,4
20	1/1 048 576	0,0000009536743	0,0001	0,95	− 120,4

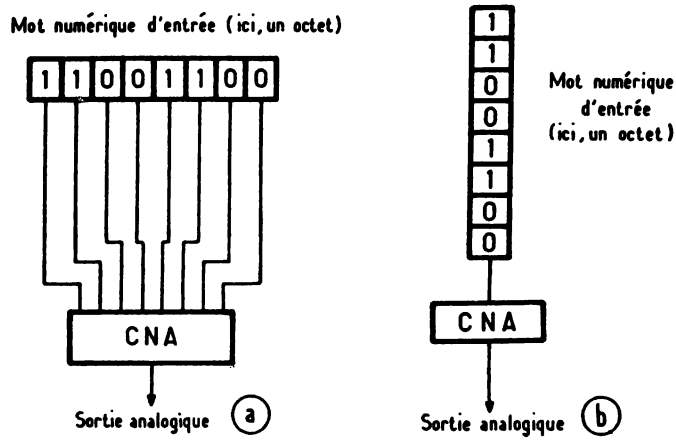


Fig. 11 — Définition des convertisseurs N/A parallèle et série.

Le nombre de bits du mot numérique que l'on applique à l'entrée définit la résolution du convertisseur (mais non sa précision). Par exemple, il est évident qu'avec un mot de 8 bits, pouvant donc coder 256 combinaisons, le bit de plus faible poids qui définit la résolution pèse 1/256 ; si l'on ramène ce 1/256 à une échelle sur 100, on trouve que 100/256 = 0,39. Par conséquent, un mot numérique sur 8 bits ne peut fournir qu'une résolution de 0,39 % ; ce n'est qu'en accroissant la longueur du mot qu'on peut augmenter la résolution, le tableau donnant quelques valeurs-types.

3.1. Convertisseur à résistances pondérées

CONVERSION A RESISTANCES PONDEREES

Le principe du convertisseur à résistances pondérées est indiqué dans la figure 12. Chaque bit du mot d'entrée commande un interrupteur qui sera fermé s'il est à 1, ouvert s'il est à 0. D'autre part, ces interrupteurs mettent en service des résistances de valeurs croissantes, d'échelle 2, toutes alimentées par la même source. Par conséquent, la résistance 2R est parcourue par un courant deux fois moindre que R (lorsque l'interrupteur correspondant est fermé) et deux fois supérieur à celui qui traverse 4R ; le même raisonnement s'applique à toute la chaîne.

Les courants élémentaires correspondent aux poids respectifs des bits (binaires) et sont sommés par un amplificateur opérationnel dont la sortie

fournit une tension analogique. Cet amplificateur peut être omis si l'on désire une sortie en courant, ou encore remplacé par une résistance.

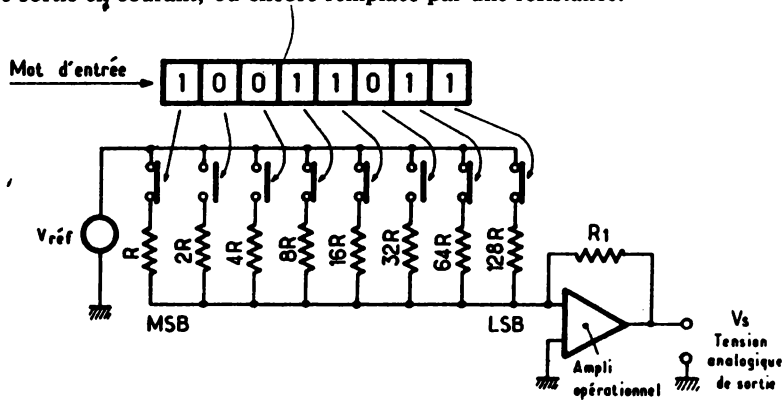


Fig. 12 — Principe du convertisseur parallèle à résistances pondérées. Le courant pondéré obtenu est additionné par un amplificateur opérationnel.

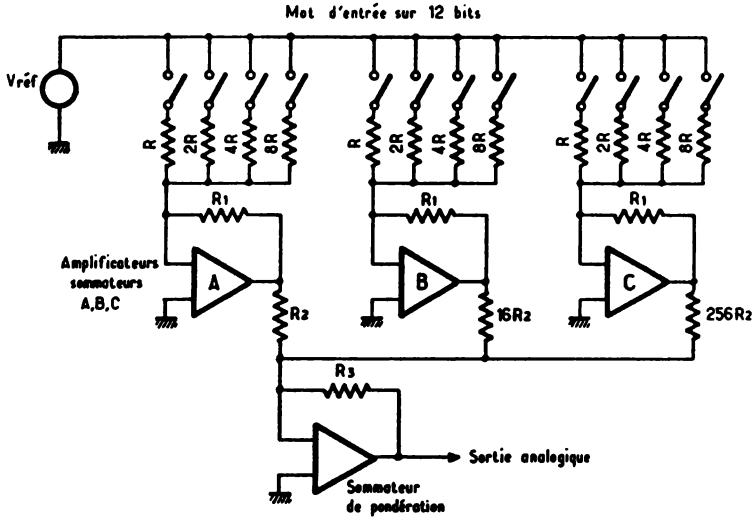


Fig. 13 — Convertisseur à réseau pondéré dans lequel, au prix d'un nombre supérieur d'amplificateurs, on réduit l'étalement des valeurs de résistances.

Ce type de convertisseur a pour lui la simplicité et une vitesse élevée. La difficulté majeure, dans sa réalisation, réside dans les valeurs largement différentes des résistances, auxquelles on demande, en outre, une grande précision et une grande stabilité. Aussi préfère-t-on généralement les regrouper par blocs de quatre valeurs seulement, la pondération étant exécutée à la sortie (fig. 13).

3.2. Convertisseurs à échelle R-2R

CONVERSION A ECHELLE R/2R

Par rapport au précédent, le convertisseur à échelle R-2R offre l'avantage de n'utiliser que deux valeurs de résistances, simple et double ; de surcroît, leur précision absolue n'offre guère d'importance car seule intervient leur précision relative (leur rapport). La version la plus usuelle de ce convertisseur est celle dite à *échelle inversée* où l'on commute des courants.

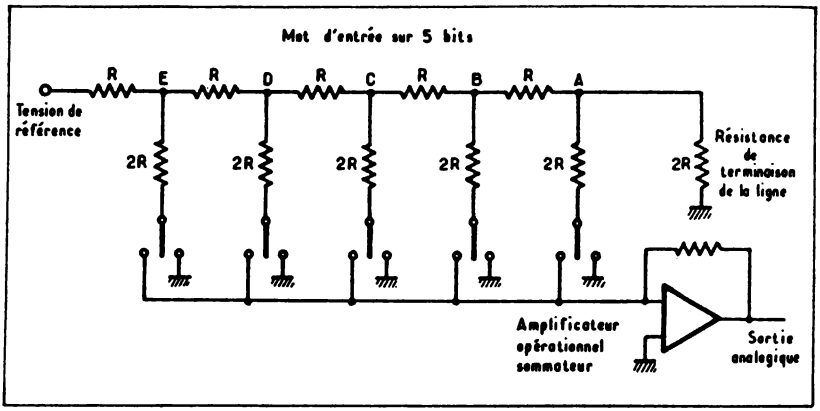


Fig. 14 — Principe du convertisseur R/2R à commutation de courants, ici pour une entrée sur 5 bits.

Pour en comprendre le fonctionnement, on va se référer au schéma de la figure 14. Chaque bit du mot d'entrée commande un commutateur connecté vers la masse pour un zéro, vers la ligne de sommation aboutissant à l'amplificateur opérationnel pour un 1 ; la tension à l'entrée de cet amplificateur opérationnel étant nulle par définition (c'est le principe même de fonctionnement de ce composant qui l'impose), on constate que, quelle que soit la position du commutateur, la résistance 2R qu'il pilote est reliée à un potentiel nul, ce qui va faciliter l'analyse du montage.

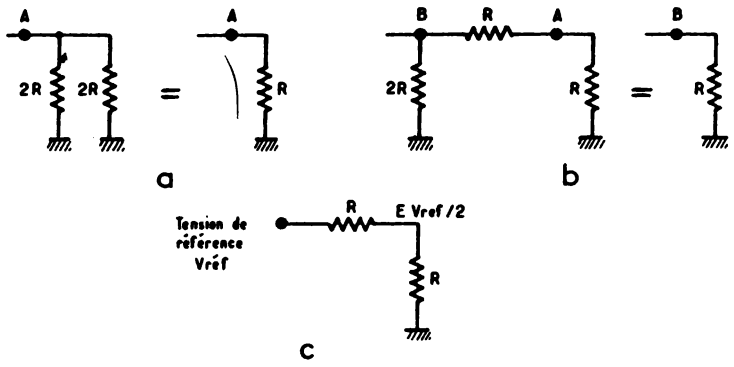


Fig. 15 — Explication du fonctionnement du convertisseur R/2R : l'association des résistances mène toujours à une résultante égale à R ou à 2R.

Ainsi, si l'on se place au point noté A et qu'on regarde vers la droite, on voit apparaître deux résistances 2R en parallèle ; l'impédance résultante est donc R, ce qu'on a rappelé dans la figure 15*a*. Reculons alors en B, toujours en observant la droite du schéma ; on voit ici une résistance R en série avec la R trouvée à partir de A, soit 2R au total, le tout en parallèle sur 2R : la résultante donne à nouveau R, comme le montre la même figure, en *b*. Le même raisonnement vaut pour C, D ou E : en quelque point qu'on se place, on voit apparaître une résistance R qui se trouvera ainsi en série avec une dernière résistance R reliée à une source de référence (en *c*). Au point E, on a donc $V_{ref}/2$; on en déduit, en développant ce raisonnement mais de la gauche vers la droite (ce qui est résumé dans la figure 16) qu'on a $V_{ref}/4$ en D, $V_{ref}/8$ en C, $V_{ref}/16$ en B et $V_{ref}/32$ en A. A travers les résistances 2R, ces tensions donnent

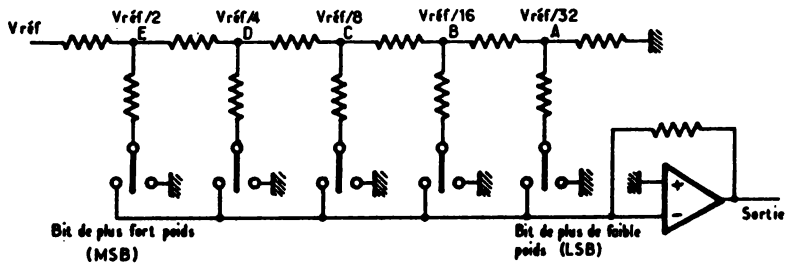


Fig. 16 — Distribution des tensions le long de l'échelle R/2R. On repère aisément la position des bits de plus fort (MSB) et plus faible (LSB) poids.

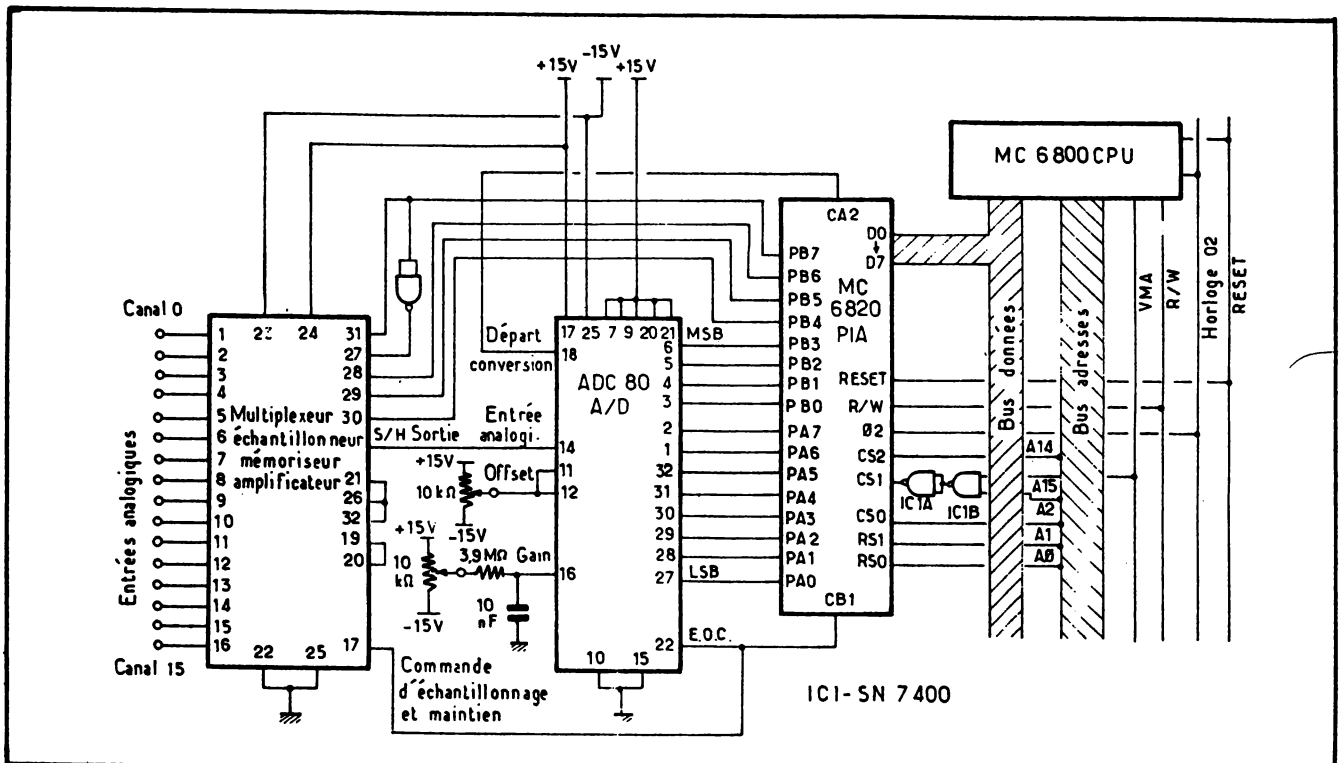


Fig. 17. — Acquisition sur 16 voies avec conversion sur 12 bits pour microprocesseur 6800.

naissance à des courants qui seront additionnés par l'amplificateur opérationnel. Dans le schéma, le mot d'entrée est sur 5 bits binaires ; il peut être allongé (ou réduit) selon le même principe ; les commutateurs sont, bien sûr, des transistors.

4. EXEMPLE DE MONTAGE A/D

Les montages de convertisseurs analogique-numérique sont aussi nombreux que les circuits proposés, multipliés par les types de microprocesseurs auxquels ils s'adressent. Le plus souvent, on aura intérêt à acquérir un circuit intégré convertisseur répondant aux spécifications de l'application et compatible avec le bus du système utilisé.

La figure 17 donne une telle application, et elle a surtout valeur d'exemple en ce sens que le convertisseur utilisé se connecte directement à un circuit d'entrées-sorties parallèles type PIA (le 6820) d'un système à microprocesseur 6800. Il peut être attaqué par une entrée analogique à évolution lente, ou précédé par le multiplexeur-échantillonneur-mémoriseur du schéma. Ce système, proposé par *Micro-Network*, accepte ainsi 16 canaux et convertit sur 12 bits les tensions d'entrée.

CIRCUIT ADC80

Les circuits nécessaires, le MN7130 qui sert de multiplexeur 16 voies (ou 8 voies différentielles) et le ADC80, convertisseur A/D sur 12 bits, sont tous deux en boîtiers DIL de 32 broches. Le système peut traiter 30 000 voies par seconde, avec un temps d'ouverture de 50 ns et une consommation de 2 W.

La gamme des tensions d'entrée est adaptable par câblage aux différents standards : ± 10 V, ± 5 V, ± ,25 V, 0 à 5 V ou 0 à 10 V. La résolution est de 12 bits, ce qui offre une excellente précision.

Dans le système utilisé ici, quatre bits d'un des deux ports du PIA sont réservés à l'adresse du canal sélectionné ; les quatre autres bits complètent les 8 du second port pour former le mot de 12 bits qui lira le résultat de la conversion. Les commandes sont programmées, en raison de la vitesse d'acquisition réalisée ; la ligne CA2 fournit l'impulsion de départ de la conversion, alors que le DB1 détecte la fin de conversion. Pour le reste, le montage est des plus classiques.

Le logiciel est, lui aussi, très simple, tout en faisant intervenir une temporisation couvrant le temps d'acquisition (une dizaine de microsecondes).

5. COMMANDES DE MANETTES DE JEUX DE L'IBM-PC

CONVERSION PAR MONOSTABLE
MANETTES DE JEUX DE L'IBM-PC

L'exemple d'un mode de conversion A/D simple est fourni par les commandes de manettes de jeux, lesquelles pourraient, d'ailleurs, intervenir pour d'autres objectifs.

Dans le cas de l'IBM-PC, une carte d'interface supplémentaire est requise, pour deux manettes de jeux (« manches à balai »). La variation de position est prélevée sur des potentiomètres de 100 kΩ, pour les deux axes. La résistance instantanée entre dans le calcul d'une constante de temps qui détermine la fréquence d'impulsions d'un monostable. La figure 18 montre la structure de principe des manettes et leurs liaisons au connecteur 15 broches. L'interface dispose de son connecteur 15 broches, type D, avec 4 entrées analogiques (résistances) et 4 entrées numériques (poussoirs), comme le montre la figure 19. En fait, ces entrées peuvent servir à tout autre usage que celui qu'indique le titre de ce paragraphe.

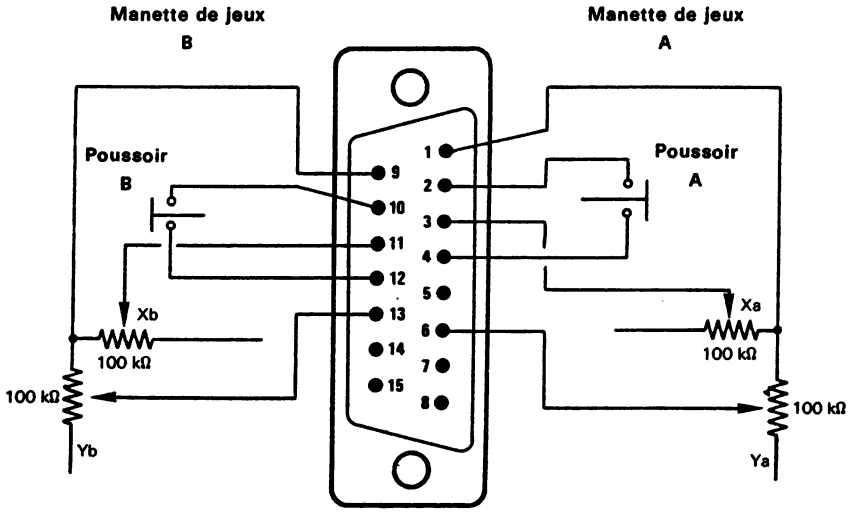
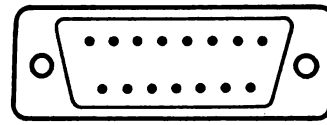


Fig. 18 — Connexion des deux manettes de jeux (IBM-PC).

A) Organisation du connecteur



B) Brochage

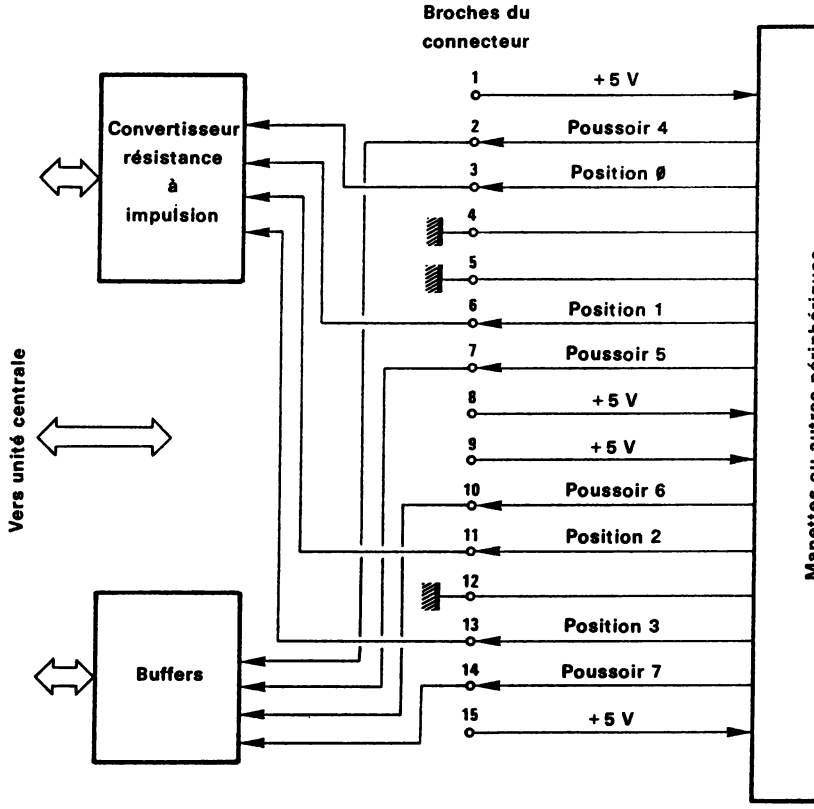


Fig. 19 — Le connecteur du panneau arrière de l'unité centrale, son organisation (A), avec le principe de la carte d'interface et les connexions (B), pour l'IBM-PC.

CHAPITRE XIII

CODES BINAIRES

Les propriétés du binaire habituel rendent difficiles certains problèmes de conception, de calcul, de fonctionnement, de synchronisation ou de détection d'erreur. Aussi a-t-on imaginé de très nombreux codes transposant le binaire, dont les principaux sont le « non retour à zéro » (NRZ), le RZ, les codes biphases et multiniveaux, les FM et MFM pour l'enregistrement sur disquettes, etc.

Les mots-clés de ce chapitre

Unipolaire : code binaire, 242
Polaire : code binaire, 242
Bipolaire : code binaire, 242
Ternaire : code ternaire, 242

NRZ : codes sans retour à zéro, 242
NRZI : code NRZ inversé, 244
RZ : code à retour à zéro, 245
Biphases : codes binaires, 245

Miller : code binaire, 248
DM : ou code Miller, 248
FM : code à modulation de fréquence, 249
MFM : code FM modifié, 249

MMFM : code MFM modifié, 251
M2FM : autre nom du code MMFM, 251
GCR : code pour enregistrement magnétique, 252
Multiniveau : codes binaires, 253

Baud : unité de débit, 254
Bits par seconde : unité de débit, 254

1. CODES UNIPOLAIRE, POLAIRE, BIPOLAIRE

CODE UNIPOLAIRE

Définissons, préalablement, la terminologie employée. Un code est dit *unipolaire* lorsque les niveaux utiles sont de même signe. Dans le cas des circuits intégrés TTL qui servent souvent de référence, ces niveaux seront de zéro volt pour le niveau *bas* logique (niveau 0) et de 3,5 V à 5 V pour le niveau *haut*, ou niveau 1.

CODE POLAIRE

CODES BIPOLAIRE, TERNAIRE

Un *code polaire* est un code dans lequel les deux niveaux sont de *signe opposé*, qu'ils soient symétriques ou non. Dans la figure 1, on a aussi représenté un code polaire asymétrique ; souvent, on dira du code polaire *symétrique* qu'il est *bipolaire*. Si, de plus, on se réfère également au niveau zéro, on aboutit à un code *ternaire*.

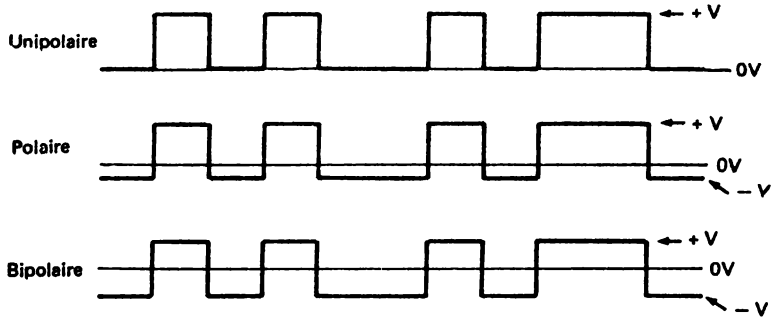


Fig. 1 — Différence entre codes unipolaire, polaire et bipolaire.

2. CODES NRZ

NRZ

Le code le plus simple à réaliser est probablement le NRZ, ces initiales provenant de *Non Retour à Zéro*. Dans ce code, le niveau *haut* à coder ne donne pas

naissance à une impulsion revenant à zéro, comme c'est le cas avec les codes RZ, ou à *Retour à Zéro* ; la figure 2 met en évidence en quoi réside la différence.

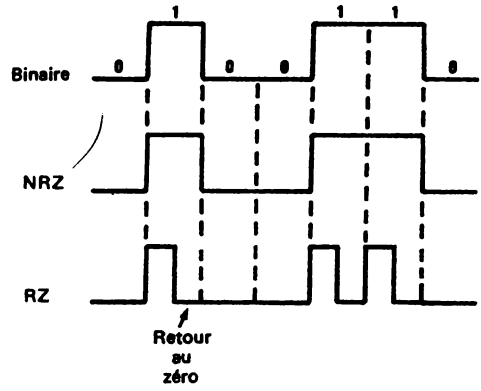


Fig. 2 — Un code NRZ (retour à zéro) revient à zéro après chaque impulsion ; ce n'est pas le cas avec le NRZ (non retour à zéro).

CODES NRZ

Il existe plusieurs codes NRZ ; ils sont simples à créer et à décoder, mais présentent une forte composante continue. Les principaux sont :

• le NRZ/L : les niveaux logiques *hauts* et *bas* sont représentés par des niveaux unipolaires ou polaires de la façon la plus simple qui soit (fig. 3) ;

• le NRZ/M : les niveaux logiques *hauts* provoquent un changement du niveau émis, quel qu'il ait été, alors que les niveaux logiques *bas* ne se traduisent par aucun changement ;

• le NRZ/S : c'est le symétrique du précédent. Un niveau *haut* ne provoque aucun changement alors qu'un niveau *bas* en déclenche un ;

• on appellera également NRZI *(« Non Return To Zero Inverted »)* un code dans lequel un changement d'état est provoqué par un changement de niveau binaire.

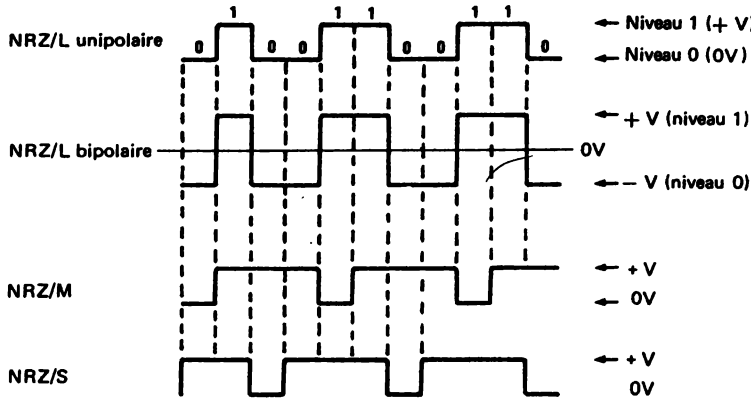


Fig. 3 — Caractéristiques de quatre codes NRZ.

2.1. Le NRZI

NRZI

Ainsi, et avec le NRZ, on travaille sur deux niveaux correspondants au 0 et au 1 binaires. La lecture d'un enregistrement NRZ peut se faire en détectant simplement un *changement de niveau* (sans lire une valeur absolue) et en positionnant en conséquence un flip-flop ; c'est souvent ainsi que l'on procède mais, malheureusement, il suffit de manquer une seule inversion pour que toute la chaîne des informations lues ensuite positionne « à l'envers » le flip-flop : c'est ce qu'on appelle *l'erreur de propagation*.

Pour l'éviter, on préfère alors employer le NRZ *inversé*, noté NRZI. Ici, un 1 provoque une inversion du niveau, quel que soit son niveau précédent (0 ou 1), un 0 laissant le même niveau (fig. 4).

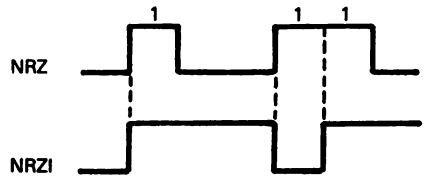


Fig. 4 — En NRZI, le niveau bascule pour chaque 1.

Ainsi, et même si l'on manque une lecture d'un 1, la chaîne qui suit n'est pas affectée : on supprime l'erreur de transmission.

3. CODES RZ

RZ

Les codes RZ, pour *Retour à Zéro*, associent une horloge aux signaux émis et sont ainsi à synchronisation intrinsèque.

PRINCIPAUX CODES RZ

• **Le RZ de base, *unipolaire*. Pour l'obtenir, on applique à une porte logique ET à deux entrées le signal binaire, du type NRZ/L, et une fréquence double d'horloge (fig. 5).**

• **Le *code à modulation de position*, appelé PPM pour « *Pulse Position Modulation* ». Les 0 et 1 se traduisent par des impulsions de 1/4 de la durée du bit, situées dans le premier quart pour les 0, le troisième quart pour les 1 (fig. 6).**

• ***Modulation de largeur d'impulsion* : on appelle PWM pour « *Pulse Width Modulation* » ou PDM, pour « *Pulse Duration Modulation* ». Ici, si l'on divise par trois la durée d'un bit, un zéro sera représenté par un niveau maintenu un tiers du temps et un 1 par un niveau s'étendant sur 2/3 (fig. 7). On remarquera que chaque flanc montant correspond au départ d'un nouveau bit, ce qui permet d'extraire aisément un signal d'horloge.**

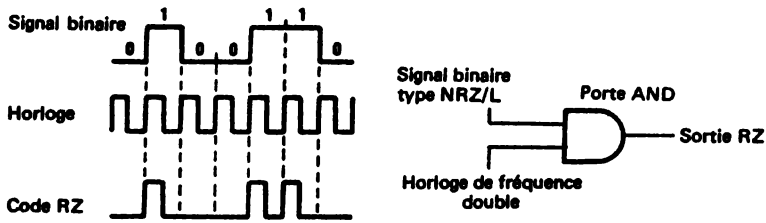


Fig. 5 — Réalisation d'un signal RZ.

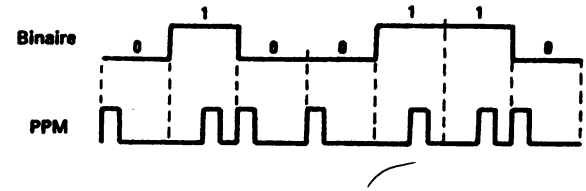


Fig. 6 — Code RZ à modulation de position (PPM).

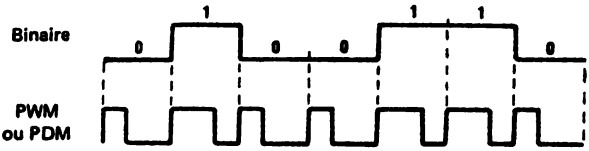


Fig. 7 — Code RZ à modulation de largeur d'impulsion.

4. CODES BIPHASES L, M ET S

BIPHASES

Avec les codes biphases, on supprime la composante continue (à l'exception d'un seul). Ils restituent aisément le rythme mais offrent parfois une certaine ambiguïté d'horloge.

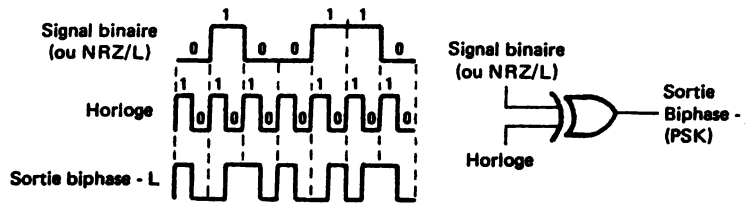


Fig. 8 — Réalisation d'un code biphase-L.

PRINCIPAUX CODES BIPHASES

• **Le BI-L, ou *code Manchester*, appelé encore PSK, pour *« Phase Shift Keying »*. Il est obtenu en appliquant simultanément à une porte OU exclusif, cette fois, le signal binaire et une horloge de période moitié de la durée d'un bit (fig. 8). On constate qu'à chaque bit, une transition se manifeste ; si deux bits qui se suivent sont de même niveau, il y a transition ; s'ils sont différents, il n'y a pas de transition « à l'interface ».**

• **Les codes FSK (pour *« Frequency Shift Keying »*), qui se subdivisent à leur tour en plusieurs variantes *complémentaires* (codes M ou codes S), ou à transition *au milieu* (code 1) ou *au début* (code 2) :**

— le biphase M-1 comporte toujours une transition à mi-bit, pas de transition au début pour les 1, une transition au début pour les 0. Etudié plus particulièrement par *TRT* sous le nom de *biphase différentiel*, il a fait l'objet d'une contribution XVII-56 du *CCITT*. On remarquera que le codage d'un zéro reprend le codage du bit précédent et celui d'un 1 son inverse (fig. 9) ;

— le biphase M-2 marque toujours une transition au début d'un bit, une transition au milieu pour les 1, une absence de transition au milieu pour les 0 ;

— le biphase S-1 offre une transition au milieu, et à l'inverse du M-1, une transition au début pour les 1 et pas de transition pour les 0 ;

— enfin, le biphase S-2 est, lui aussi, l'inverse du M-2 avec une transition obligatoire au début mais à mi-bit, pas de transition pour les 1.

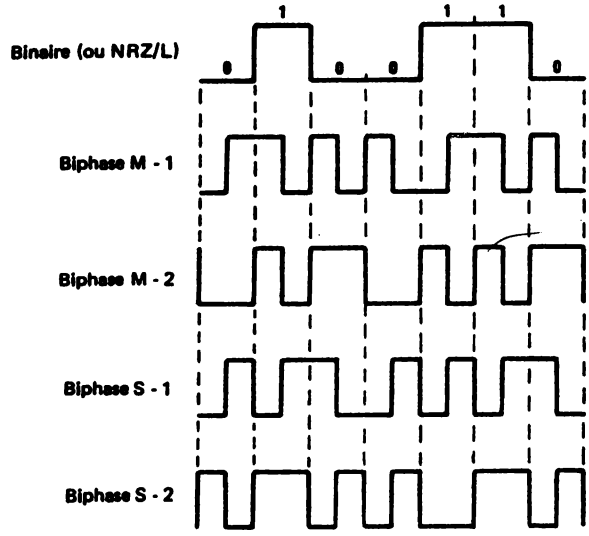


Fig. 9 — Quatre codes biphases.

5. CODES MILLER (DM)

MILLER
DM

Les codes à *modulation de délai* (DM), appelés aussi *codes Miller*, ou codes MPDC *(« Modified Phase Delay Coding »)* font également partie des codes biphases. Ils existent en plusieurs variantes, dont le DM fondamental, sa variante modifiée DM/A et celle à bande réduite DM/BR, le Manchester modifié ou biphase différentiel (BI/DI), le code par opposition de phase modifié (MPSK)...

Dans le DM, un 1 se traduit par une transition à mi-bit alors qu'un zéro ne donne lieu à une transition en fin de bit que si le bit suivant est un zéro (fig. 10).

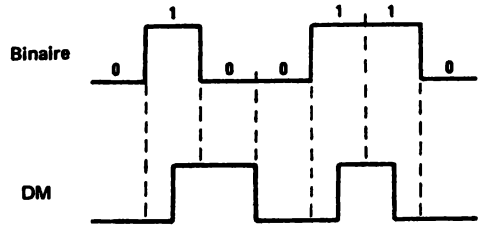


Fig. 10 — Code à modulation de délai (DM).

6. CODES FM ET MFM

FM
MFM

Les codes FM *(modulation de fréquence)* et MFM *(modulation de fréquence modifiée)* sont très utilisés pour l'enregistrement sur disquette :

- le FM pour les disquettes simple densité ;
- le MFM pour les double densité (cas des disquettes de l'IBM-PC).

En FM, les temps sont divisés en *sections* correspondant d'ailleurs à des longueurs de pistes sur les disquettes et telles qu'une section représente 4 µs d'enregistrement sur une disquette 8" et 8 µs sur une 5"1/4. Chaque section comporte d'abord un signal d'horloge, puis le bit du signal, binaire. La figure 11 montre l'aspect du signal ; on a noté le signal d'horloge H, et D le signal de donnée à 1. On constate que la fréquence d'impulsions varie du simple pour le Ø au double pour le 1, d'où le nom de modulation de fréquence. Un changement de flux a donc lieu dans les têtes (lecture ou écriture) au moins une fois par 8 µs pour une disquette 5"1/4, soit une vitesse de transfert de 125 kilobits par seconde. Chaque impulsion provoque une inversion du courant d'écriture et, partant, un basculement du sens de l'aimantation sur la disquette.

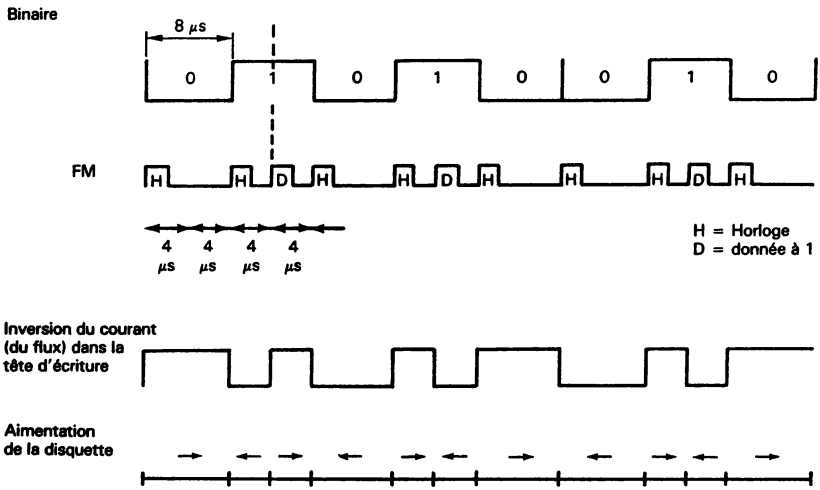

Binaire
8 µs
0 1 0 1 0 0 1 0
FM
H H D H H D H H H D H
4 µs 4 µs 4 µs 4 µs
H = Horloge
D = donnée à 1
Inversion du courant (du flux) dans la tête d'écriture
Aimentation de la disquette

Fig. 11 — Principe du codage FM et aimantation résultant de la couche magnétique sur la disquette.

Les circuits de codage et de décodage restent très simples puisqu'il suffit de générer ou de détecter des impulsions de synchronisation puis, 4 secondes plus tard (pour une disquette 5''1/4), l'impulsion de signal. La figure 12 montre comment, à la lecture et après extraction de l'horloge, on reconstitue le message initial. Par contre, ce principe est très inefficace car l'horloge occupe la moitié des temps (et espaces sur disque) ; aussi a-t-on imaginé la MFM.

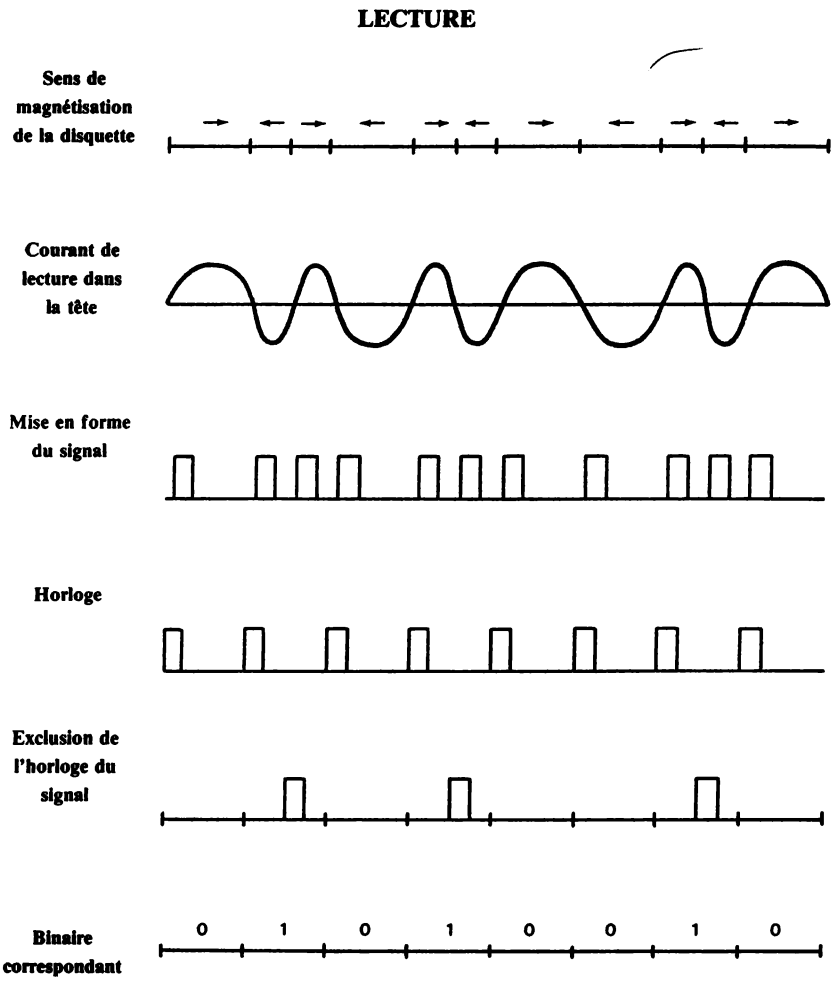

LECTURE
Sens de magnétisation de la disquette
Courant de lecture dans la tête
Mise en forme du signal
Horloge
Exclusion de l'horloge du signal
Binaire correspondant
0 1 0 1 0 0 1 0

Fig. 12 — Reconstitution du message lu, à partir d'un enregistrement en FM.

En MFM, on n'inscrit qu'une impulsion maximum par section, horloge *ou* bit de donnée à 1. En outre, le bit d'horloge ne viendra dans une section à Ø (pour la donnée) que *si* rien n'a été inscrit dans la cellule précédente (qui doit donc être à zéro, elle aussi), ce que montre la figure 13. Les impulsions d'horloge sont placées en début des sections, celles de données au milieu. On constate une forte réduction du nombre d'impulsions ; elle permet de réduire de moitié les temps, donc de porter la durée d'une section à 4 µs (disquettes 5''1/4), autrement dit, de doubler la densité. La lecture s'effectue sur les mêmes principes qu'avec la FM.

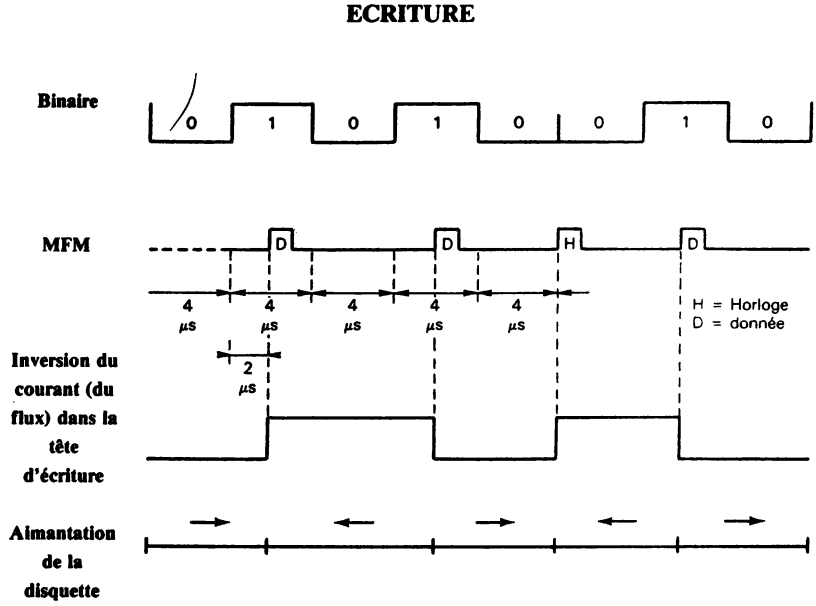

ECRITURE
Binaire
0 1 0 1 0 0 1 0
MFM
D D H D
4 µs 4 µs 4 µs 4 µs 4 µs
2 µs
H = Horloge
D = donnée
Inversion du courant (du flux) dans la tête d'écriture
Aimantation de la disquette

Fig. 13 — Principe du codage MFM, utilisé en particulier pour l'enregistrement sur disquettes double densité, cas des disquettes de l'IBM-PC.

MMFM
M²FM

Une variante de la MFM, appelée le *MFM modifiée*, ou MMFM (ou encore, M²FM), va jusqu'à supprimer les impulsions d'horloge consécutives, pour des chaînes de zéros, n'en conservant que la première (donc, celle du second zéro). En MFM tout comme en M²FM, une section est de 4 µs et, par conséquent, la vitesse de transfert des informations passe à 250 kilobits par seconde.

7. LE CODE GCR

GCR

Afin d'accroître les capacités d'enregistrement, il faudrait parvenir à supprimer purement et simplement les impulsions d'horloge. C'est ce que permet le code GCR *(« Group Coded Recording »)*, déjà utilisé pour l'enregistrement sur bande. Le principe est le suivant :

- chaque octet de donnée à écrire est divisé en 2 quartets ;
- chaque quartet est traduit par 5 bits (binaires).

A la lecture, on procèdera à l'opération inverse. Le codage sur 5 bits exige que :

- on dispose d'au moins une inversion de flux ;
- il n'y aura jamais plus de 2 zéros d'affilée.

Or, 5 bits permettent 32 combinaisons binaires ; sur ces 32, seules 16 répondent à cette dernière condition ; ce sont elles qui ont été retenues (voir tableau).

LE CODE GCR

Hexadécimal	Binaire	Code GCR
0	0000	11001
1	0001	11011
2	0010	10010
3	0011	10011
4	0100	11101
5	0101	10101
6	0110	10110
7	0111	10111
8	1000	11010
9	1001	01001
A	1010	01010
B	1011	01011
C	1100	11110
D	1101	01101
E	1110	01110
F	1111	01111

Avec le code GCR, la durée d'une section est ramenée à 2,6 µs (fig. 14) ; la vitesse de transfert passe à 380 kilobits par seconde. L'intervalle maximal entre 2 impulsions est de 7,8 µs.

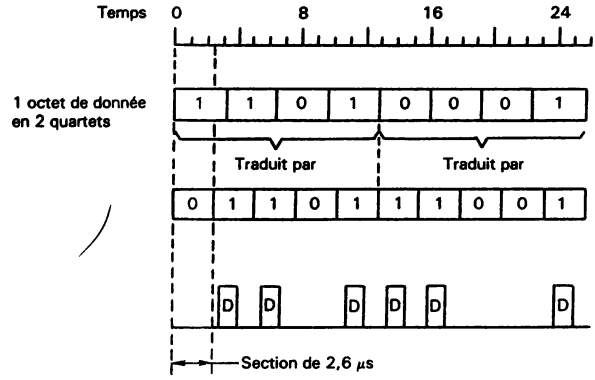


Fig. 14 — Principe du codage GCR permettant des vitesses de transfert de 380 k bits/s.

8. CODES MULTINIVEAUX

MULTINIVEAUX

Les codes multiniveaux, ou MLC *(« Multi Level Codes »)* qu'on va examiner utilisent trois niveaux : le zéro, et + V et − V symétriques. On va voir qu'il ne s'agit pas, ici, de codes ternaires ; aussi les appelle-t-on parfois de *pseudo-ternaires.* Les principaux sont :

• le *retour à zéro,* où le 1 se traduit par une impulsion positive et le zéro par une impulsion négative, durant dans les deux cas un demi-bit (fig. 15) ;

• le *code à inversion d'impulsions* (BN-NRZ ou AMI) : les zéros ne donnent naissance à aucune impulsion, alors que les 1 créent des impulsions de polarité alternée ;

• les *codes* : bipolaire RZ, le dicode, etc. On en énumère en effet plus d'une dizaine.

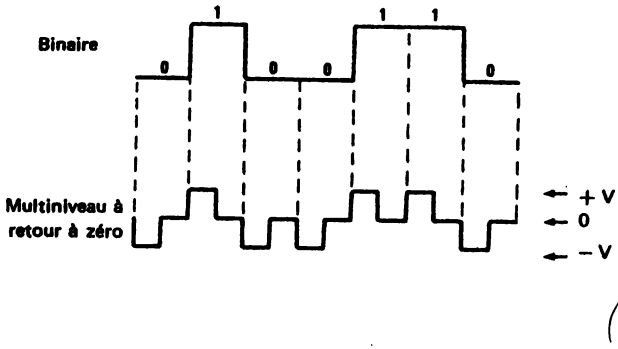


Fig. 15 — Exemple de code multiniveau (mais non ternaire).

9. BITS PAR SECONDE ET BAUDS

A propos de ces codes numériques, arrêtons-nous un instant sur des définitions fondamentales. Il ne faut pas confondre *bits par seconde* et *bauds*, car ces deux unités de mesure de débit de transmission d'informations numériques ne sont pas synonymes :

BAUD
BITS/S

• Un débit mesuré en *bauds* indique le nombre d'*états discrets* ou de *niveaux transmissibles* en une seconde.

• Un débit mesuré en *bits par seconde* indique le nombre de bits transmis pendant l'unité de temps, c'est-à-dire le nombre de niveaux 0 ou 1 car il n'y en a alors que deux possibles.

Pour bien montrer la différence, examinons le cas d'une transmission par groupes de deux bits, appelés *dibits*. Il faudra 4 niveaux de tension pour coder les quatre combinaisons possibles (fig. 15) :

Niveau 0 → 00
Niveau 1 → 01
Niveau 2 → 10
Niveau 3 → 11

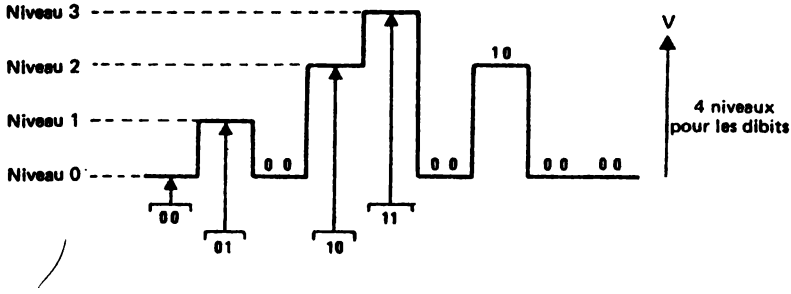


Fig. 16 — Quatre niveaux de tension servent à coder des dibits.

Si, en une seconde, on transmet deux niveaux de tension, on aura transmis *quatre bits*. La vitesse est alors de :

- 4 bits par seconde, ou de :
- 2 bauds.

En effet, si l'on transmet, par exemple, les débits de niveaux 2 et 3, on aura transmis le mot binaire 1011, soit un quartet.

Supposons qu'en transmission série asynchrone, on travaille avec des caractères ASCII codés sur 10 bits et à la vitesse de 1 200 *bits par seconde*. Chaque bit occupe alors 833 microsecondes, et un caractère 8,33 ms. En transmettant des dibits avec la même cadence, soit 1 200 par seconde, et donc *en travaillant toujours à 1 200 bauds*, chaque caractère n'occupera plus que 4,165 ms. On en transmettra deux fois plus dans le même temps, soit 2 400 bits/seconde.

La figure 17 montre comment le même message peut ainsi être émis en 16 coups d'horloge en binaire, et en 8 coups seulement en dibits ; dans le premier cas, le débit est deux fois moindre.

Des configurations autres que les dibits existent. En conclusion, on ne peut donc confondre bauds et bits/seconde que lorsque la liaison est assurée en binaire sur deux niveaux.

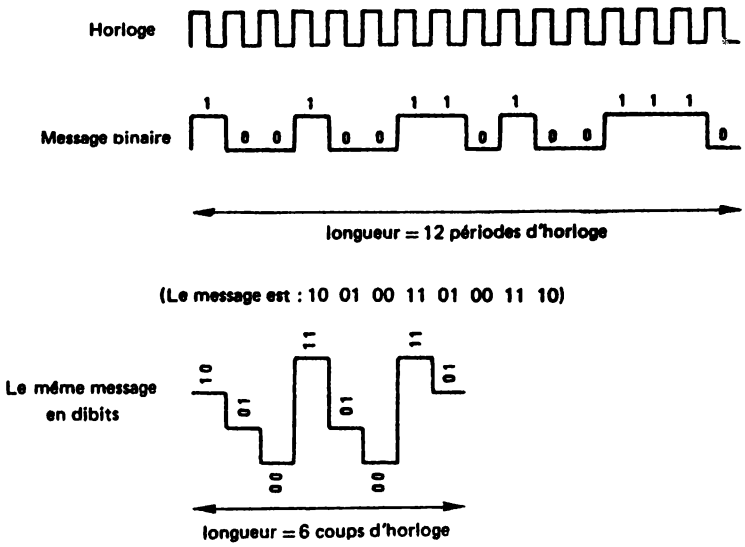

Horloge
Message binaire
1 1 0 0 1 1 0 0 1 0 1 0 0 1 1 1 0
longueur = 12 périodes d'horloge
(Le message est : 10 01 00 11 01 00 11 10)
Le même message en dibits
10 01 00 11 01 00 11 10
longueur = 6 coups d'horloge

Fig. 17 — Comparaison entre l'émission d'un même message en binaire et en dibits.

Exemples de calcul du débit en bauds

Le débit, en bauds, est l'inverse de la durée (en secondes) du plus court signal transmis (qu'il soit appelé *marque*, *espace*, 0 ou 1). Si la durée est de 20 ms, soit 0,02 s, le débit sera :

1/0,02 = 50 bauds,

ce qui est le débit caractéristique du télex.

CHAPITRE XIV

DÉTECTION ET CORRECTION D'ERREURS

On dit qu'une erreur entache un message binaire lorsque le mot reçu n'est pas rigoureusement identique à celui qui a été émis. Pour détecter des erreurs, plusieurs méthodes ont été mises au point, certaines d'entre elles pouvant même déclencher une correction automatique.

Les mots-clés de ce chapitre

Erreurs : détection et correction, 256
Retransmission : pour corriger les erreurs, 257
Emission doublée : pour corriger les erreurs, 257
Echo : mode de détection d'erreurs, 257

ARQ : mode de détection des erreurs, 258, 274
FEC : mode de détection et correction des erreurs, 258
Parité : d'un mot binaire, 258
Imparité : d'un mot binaire, 258

Parité transversale, 259
Parité verticale, 259
Parité longitudinale, 261
Codes 2/5 : pour détecter les erreurs, 263

Double complémentation : pour situer une erreur, 265
Hamming : codes de détection d'erreurs et de correction, 266
Codes cycliques redondants : pour détecter les erreurs, 269
CRC : code cyclique redondant, pour détecter les erreurs, 269

Polynominaux : codes de détection d'erreurs, 273
ACK : accusé de réception positif, 274
NACK : accusé de réception négatif, 274
Circuit DP 8400 : de détection et correction d'erreurs, 275

1. LES ERREURS ET LEUR TRAITEMENT

Le message binaire que l'on reçoit doit rester rigoureusement conforme à celui qui a été émis. Or, des bruits divers peuvent entacher la liaison et transformer des 0 en 1 ou des 1 en 0 ; leurs sources résident dans les parasites, des bruits, des transitoires de commutation, etc. En outre, certains composants du système peuvent être défectueux, ou des connexions mal établies, ce qui crée autant de sources d'erreurs.

En pratique, on classera souvent les erreurs en deux catégories :

ERREURS (CORRECTION)

a) Les *erreurs matérielles*, ou *hard*, dues aux composants, par exemple un circuit mémoire défectueux. L'erreur est, ici, *reproductible*.

b) Les *erreurs logicielles*, ou *soft* : ce sont des erreurs aléatoires, non reproductibles. Elles peuvent être provoquées par des phénomènes parasites, des rayonnements ionisants (une particule alpha frappant un circuit mémoire et induisant un bit erroné, par exemple : on estime que le taux d'erreur aléatoire d'une mémoire RAM de 64 K bits est de l'ordre de 10^{-6} par heure).

Le tableau suivant montre que, relativement aux mémoires dont la densité ne cesse de s'accroître, par circuit comme par ordinateur, ce sont les erreurs logicielles qui prédominent.

Les erreurs dans les mémoires RAM

Capacité du circuit mémoire (K bits)	Taux d'erreur typique en %/1000 h	
	Logicielles (soft)	Matérielles (hard)
1	0,001	0,0001
4	0,02	0,002
16	0,10	0,011
64	0,13	0,016

Source : Motorola

Ainsi, si l'on considère une mémoire centrale de 16 M octets constituée de circuits de 64 K bits, le *temps moyen entre erreurs*, sans circuit de détection et correction d'erreur, sera donné ainsi :

Mémoire de 16 M octets avec circuits de 64 K bits

Type d'erreur	Temps (jours)
Logicielle, simple	13,5
Matérielle, simple	109,5
Logicielle, double	864
Matérielle, double	7 021

Ces tableaux montrent que ce sont les erreurs simples (uniques) qui prédominent, et surtout les erreurs aléatoires, logicielles. Ajoutons que l'utilisation de circuits de détection et correction d'erreurs, tels les MC6 8540 de *Motorola* ou DP 8400 de *National Semiconductor* portent le temps de 13,5 jours à plus de deux ans... On va en examiner les principes.

PRINCIPE DE LA RETRANSMISSION

Lorsque le message est très redondant, et par exemple lorsqu'il s'agit de la transmission d'un texte vocal, on peut purement et simplement ignorer ces erreurs, le correspondant rectifiant par lui-même si nécessaire. Sinon il pourra demander une *retransmission* du même message ; un tel principe peut être transposé à des systèmes automatiques dès lors qu'une erreur est détectée, on va voir comment.

EMISSION DOUBLEE

Une autre méthode applicable lorsque les échanges sont lents (enregistrement sur cassette magnétique, par exemple) consiste à répéter systématiquement le message puis à en comparer les deux versions afin de détecter ses erreurs. Plus onéreuse, la méthode consistant à le tripler puis, à l'aide d'une logique de décision majoritaire, à en extraire les termes les plus fiables, n'est guère employée.

MODE ECHO

Par contre, le principe de la ré-émission *en écho* a été, lui, souvent appliqué. Il consiste à faire réémettre le message *par le destinataire*, en écho ; l'émetteur le reçoit et compare la version d'origine avec l'écho, et cela carac-

tère par caractère. Si une erreur est détectée, le caractère est ré-émis avec l'information nécessaire pour le destinataire. On notera que, malheureusement, l'erreur peut fort bien ne survenir ici que sur l'écho.

MODE ARQ

On gagne en vitesse si la demande de ré-émission ne porte que sur le caractère erroné. Il faut alors que le destinataire puisse tester les caractères au fur et à mesure de leur réception ; pour cela, des bits supplémentaires leur sont adjoints et *leur rôle se borne uniquement au test.* Cette méthode a reçu le nom de ARQ, pour « *Automatic Request for Repeat* », soit *demande de répétition automatique.*

MODE FEC

Une dernière méthode, plus puissante, consiste non seulement à détecter les erreurs à l'aide des bits redondants, mais encore à se servir d'eux pour les corriger. C'est la FEC, pour « *Forward Error Correction* ». Il n'y a plus de retransmission mais correction *en temps réel* par le récepteur.

En pratique et le plus souvent, on rencontrera trois types de codes de détection d'erreurs qui sont :

1) *les codes de parité*, ou *d'imparité* ;

2) *les codes redondants cycliques* (« CRC ») ;

3) *les codes de Hamming.*

2. LA PARITÉ TRANSVERSALE

PARITE, IMPARITE

On dit d'un mot binaire qu'il est pair lorsqu'il contient un nombre pair de 1. Par exemple, 01001110 est pair, alors que 01001100 est impair. La *parité* est la qualité d'un mot pair, et l'*imparité* celle d'un mot

impair ; cependant, on trouvera fréquemment les expressions de « parité *paire* » et « parité *impaire* ».

La détection des erreurs par la méthode de la parité est très utilisée lorsqu'on travaille en code ASCII. Dans ce cas, un mot se compose de 7 ou 8 bits ; si l'on veut aboutir à un octet, on dispose d'un bit supplémentaire qui pourra servir à imposer des mots successifs *uniquement pairs*, ou *uniquement impairs* selon la convention fixée au préalable. Prenons des exemples.

Supposons que l'on recherche la *parité* (des mots pairs). Si le mot ASCII est, sur 7 bits, impair à l'origine :

0101010

on lui ajoutera un 1 en 8e position, celle du bit de plus fort poids, ce qui donnera :

10101010

Le mot sur un octet est bien devenu pair. A l'inverse, si le mot d'origine est pair, c'est un zéro qu'on lui ajoutera, toujours en bit de plus fort poids. Ainsi, au mot pair sur 7 bits tel que 1101010, on ajoutera un zéro qui donnera 01101010, mot pair également.

Le bit supplémentaire est calculé et ajouté à l'émission. A la réception, on vérifie que tous les mots sont pairs ; si un mot impair survient, c'est qu'une erreur s'est produite en cours de liaison et que le mot est faux.

PARITE TRANSVERSALE
PARITE VERTICALE

On pourrait tout aussi bien convenir que les mots seront tous impairs. Dans ce cas, on ajoute des 0 ou des 1 pour imposer l'*imparité*. Ainsi, à 0101010, on ajoutera un zéro pour obtenir 00101010, et à 1101010, on ajoutera un 1 pour obtenir 11101010, mots impairs sur 8 bits.

Cette parité (ou imparité) est dite *transversale* (ou parfois *verticale*), car il existe une parité *longitudinale* qui la complète, on va comprendre pourquoi plus loin.

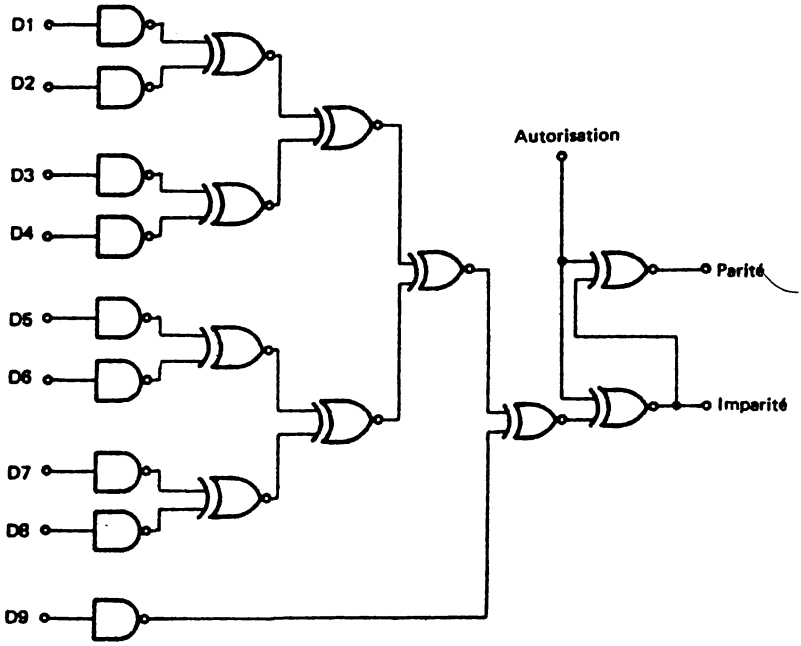


Fig. 1 — Logique de création et de contrôle de parité ou d'imparité.

La logique de création et de vérification de la parité est simple. Elle est illustrée figure 1 pour un mot de 9 bits, ce schéma étant, en fait, celui du circuit intégré spécialisé 8282 de *Signetics*. Le mot d'entrée est D_1 à D_9 ; la parité est détectée en 4 paires de (deux) bits au premier niveau, les trois niveaux suivant et des inverseurs fournissant l'information de parité ou d'imparité à la sortie. Celle-ci pourra servir de vérification, ou à l'émission, être ajoutée aux mots émis. Les entrées, prévues pour 9 bits, sont mises à la masse lorsqu'elles sont inutilisées.

3. PARITÉ LONGITUDINALE

PARITE LONGITUDINALE

La parité transversale permet de détecter un *nombre impair* d'erreurs qui entachent un mot, mais malheureusement, des erreurs paires s'annulent. Par exemple, soit en parité *paire* l'émission du mot

01101010

Si l'on reçoit 00101010, donc un mot impair, c'est qu'*une* erreur est survenue. On peut donc déclencher la procédure prévue. Mais si deux erreurs se produisent et si l'on reçoit

00001010

le mot reste pair et l'on ne détecte plus aucune erreur. Pour pallier cette difficulté, on a imaginé une *parité longitudinale* qui complète la précédente. Supposons qu'on travaille en *imparité* (la démonstration serait semblable en parité) et qu'on émette 4 mots impairs dans lesquels le bit de plus fort poids, le 8^{e}, est le bit d'imparité. Alignons ces mots l'un sous l'autre et forgeons un 5^{e} mot qui, pour ses 7 bits de plus faible poids, crée une imparité *longitudinale* ; son 8^{e} bit sera le bit d'imparité de ce mot. On obtient, par exemple :

1er mot	00101010
2^{e} mot	10011101
3^{e} mot	10000110
4^{e} mot	01100101
Test	10101011

Le dernier mot, *test*, ou *somme de test*, décèlera toute double erreur dans l'un des quatre mots qui, autrement, serait passée inaperçue. On voit en effet que, si le premier mot est devenu 01001010 (au lieu de 00101010), l'addition (verticale) est fausse.

Malheureusement, des erreurs superposées, paires, rendront à nouveau ce mode de détection inefficace. De tels cas sont cependant bien moins probables.

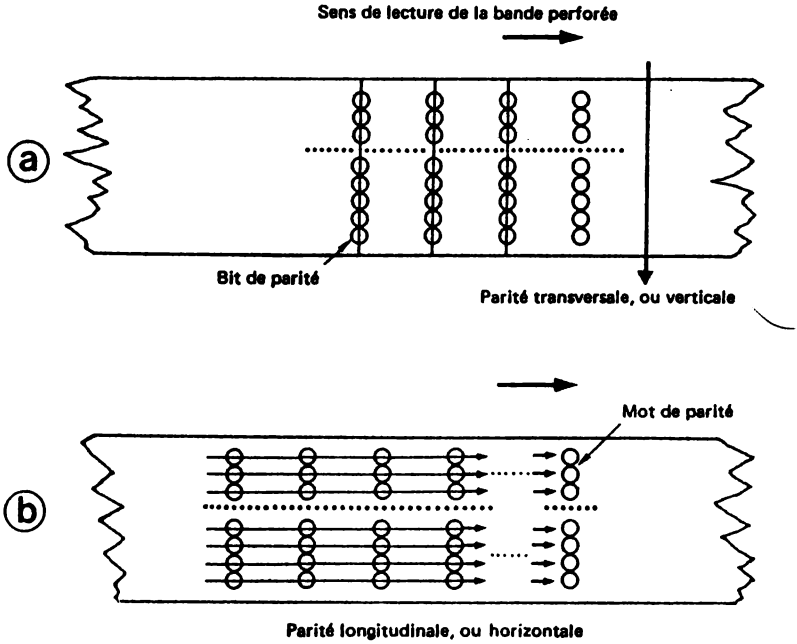


Fig. 2 — La disposition des perforations sur une bande perforée explique l'origine des appellations de parité transversale, ou longitudinale.

Remarquons que cette parité est appelée *longitudinale* ; c'est seulement la parité *transversale* qui est aussi appelée *verticale*. Cela provient simplement de la façon de présenter les mots, l'usage des bandes perforées dessinées *horizontalement* ayant entraîné de telles définitions (fig. 2).

4. UTILISATION DES CODES 2/5

CODES 2/5

Le code binaire est un code pondéré, dans lequel chaque bit dispose de son poids propre. Il est, ainsi, facile de convertir un chiffre décimal en binaire. Par contre, le binaire ne se prête pas d'emblée à un contrôle de parité : il faut ajouter un bit spécial à chaque mot.

A l'inverse, certains codes offrent directement la parité, sans bit supplémentaire. Ainsi, le code à « 2 parmi 5 » : les chiffres décimaux de 0 à 9 sont codés sur 5 bits dont 2 sont à 1 et 3 à zéro :

Code à 2 parmi 5

Décimal	En 2/5
0	1 1 0 0 0
1	0 0 0 1 1
2	0 0 1 0 1
3	0 0 1 1 0
4	0 1 0 0 1
5	0 1 0 1 0
6	0 1 1 0 0
7	1 0 0 0 1
8	1 0 0 1 0
9	1 0 1 0 0

Comme on peut le constater, le nombre de digits à 1 est toujours égal à 2. Le code biquinaire offre la même caractéristique : il est sur 7 bits, parmi lesquels deux sont toujours à 1 :

Code biquinaire

Décimal	Biquinaire : poids						
	5	0	4	3	2	1	0
0	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0
2	0	1	0	0	1	0	0
3	0	1	0	1	0	0	0
4	0	1	1	0	0	0	0
5	1	1	0	0	0	0	0
6	1	0	0	0	0	1	0
7	1	0	0	0	1	0	0
8	1	0	0	1	0	0	0
9	1	0	1	0	0	0	0

Dans ce code, chaque bit se voit attribuer un « poids » et on trouve deux bits affectés d'un zéro. De 0 à 4, le bit de poids le plus fort (le plus à gauche) est à zéro ; il passe à 1 de 5 à 9. On remarquera la symétrie de ce code, pour les bits de poids 0 à 4.

Le biquinaire comme le code à 2 parmi 5 permettent ainsi une vérification immédiate de la parité transversale. La parité longitudinale sera vérifiée en créant un mot supplémentaire. Par exemple, soit à transmettre le nombre 3587 en code à 2 parmi 5. On écrira :

3 = 00110
5 = 01010
8 = 10010
7 = 10001

Mot de parité = 01111

Le mot de parité sera donc, ici, 01111.

5. CORRECTION PAR DOUBLE COMPLÉMENTATION

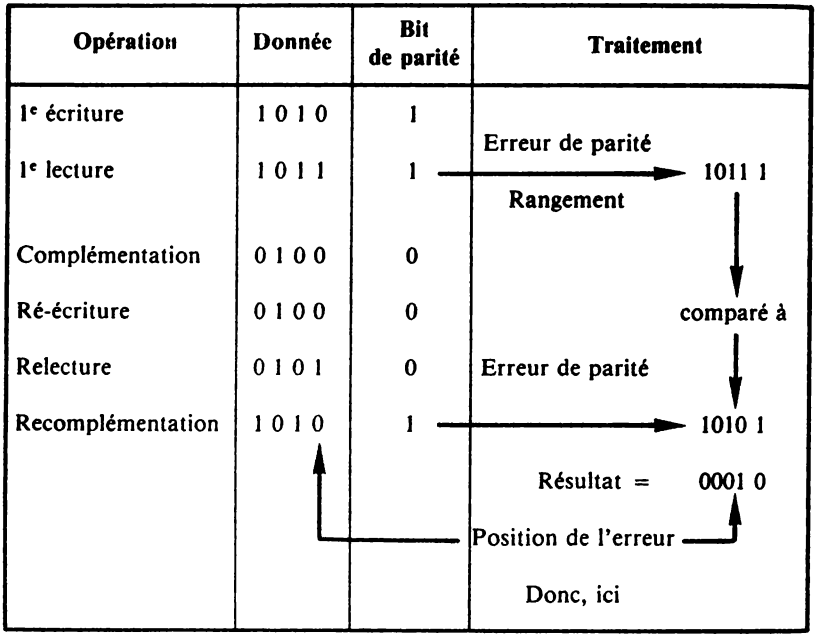
DOUBLE COMPLEMENTATION

Le simple contrôle de parité peut servir à préciser où se trouve l'erreur s'il s'agit d'une cause « matérielle », un circuit mémoire défaillant, par exemple, car l'erreur est ici *reproductible*. L'une des méthodes est la suivante :

- soit un mot de 4 bits, le mot 1010 ;
- on ajoute un 5e bit de parité, pour obtenir un nombre impair de 1 ; donc, on crée le mot 10101 ;
- on enregistre 10101 en mémoire ;
- puis, on lit ; supposons qu'on lise 10111 ;
- la vérification de la parité fait apparaître une erreur ;
- on complémente une première fois le mot *lu* et l'on obtient 01000 ;
- ce mot est inscrit au même emplacement mémoire ;
- puis il est relu : supposons que l'on relise 01010, la mémoire ayant introduit *la même erreur au même endroit*, un 1 en seconde position ;
- on complémente une nouvelle fois ce mot et l'on obtient 10101 ;
- on compare ce dernier à la première lecture, à l'aide d'une porte OU exclusif : elle fournira un 1 à l'emplacement de l'erreur.

Un tel processus, qui exige peu de moyens supplémentaires, s'applique surtout aux terminaux économiques, aux systèmes de traitement de texte, là où le temps importe peu. On retrouvera ce principe avec des circuits rapides, associés à des systèmes de vérification et correction automatique des erreurs.

Voici comme ce processus se résume :



6. CODES DE HAMMING

CODES DE HAMMING

Les codes de Hamming sont des codes détecteurs et correcteurs d'erreurs simples, pouvant être étendus toutefois à des erreurs multiples. Le principe consiste à affecter à un mot de donnée un certain nombre de bits de parité, chacun de ceux-ci surveillant un groupe précis de bits de la donnée. Dans ces codes, des bits supplémentaires sont distribués dans les mots binaires et les positions qu'ils occupent sont telles que les divers contrôles de parité conduisent à un nombre binaire *indiquant la position de l'erreur* dans le mot. Par conséquent, on inclut des bits qui allongeront le mot sans autre but que de pourvoir à la détection des erreurs, puis à leur correction.

Sans en développer la théorie, prenons un exemple à partir du code à « excès de 3 » ; c'est un code binaire dans lequel les chiffres décimaux sont transcrits avec 3 en plus :

Code à excès de 3

Décimal	Excès 3
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

On va ré-écrire ce code en plaçant les bits dans les colonnes 3, 5, 6 et 7 d'un mot à 7 bits, numérotés de 1 à 7 de la gauche vers la droite ; puis, et dans la colonne 4, on placera un bit de telle sorte que le nombre de 1 dans les colonnes 4, 5, 6 et 7 soit pair (attention, la colonne 3 n'entre pas, ici, en ligne de compte).

On obtient :

Décimal	1	2	3	4	5	6	7
0			0	0	0	1	1
1			0	1	1	0	0
2			0	0	1	0	1
3			0	0	1	1	0
4			0	1	1	1	1
5			1	0	0	0	0
6			1	1	0	0	1
7			1	1	0	1	0
8			1	0	0	1	1
9			1	1	1	0	0

Puis, on va remplir la colonne 2 de façon que le total des 1 pour les positions 2, 3, 6 et 7 soit pair. On en fait ensuite autant pour les colonnes 1, 3, 5 et 7 en remplissant la colonne 1, ce qui donne finalement :

Décimal	1	2	3	4	5	6	7
0	1	0	0	0	0	1	1
1	1	0	0	1	1	0	0
2	0	1	0	0	1	0	1
3	1	1	0	0	1	1	0
4	0	0	0	1	1	1	1
5	1	1	1	0	0	0	0
6	0	1	1	1	0	0	1
7	1	0	1	1	0	1	0
8	0	1	1	0	0	1	1
9	0	1	1	1	1	0	0

C'est donc ce code qu'on va employer. Pour vérifier son fonctionnement, supposons qu'on reçoive le mot 110011 : comporte-t-il une erreur ? L'analyse de ce mot passe par les trois étapes suivantes :

Colonnes	1	2	3	4	5	6	7	
Mot reçu	1	1	1	0	0	1	1	
Contrôle des données								
1, 3, 5, 7	1		1		0		1	Erreur 1
2, 3, 6, 7		1	1			1	1	Correct 0
4, 5, 6, 7				0	0	1	1	Correct 0

Les colonnes contrôlées 1, 3, 5, 7 donnent une somme impaire de 1 ; il y a par conséquent une erreur, notée 1. Les deux autres contrôles fournissent des résultats corrects, notés 0. A partir de ces trois résultats (0, 0 et 1), on forme un mot appelé mot de *syndrome* et qui s'écrit donc 001 : *il définit le numéro de la colonne dans laquelle se trouve l'erreur*, en binaire, donc la colonne 1, ici. Pour corriger, il suffira de complémenter le bit, c'est-à-dire de l'inverser. Puisqu'il s'agissait d'un 1 reçu, on le transformera en 0 et l'on obtiendra le mot correct, soit 0110011.

Prenons un second exemple : le mot reçu est 0110110. Les trois contrôles donnent des erreurs, donc le mot de syndrome est 111, ce qui correspond à 7,

en décimal. L'erreur réside dans la colonne 7 : le 0 doit être remplacé par 1 ; le mot est donc réellement 0110011.

Avec cette méthode, le nombre de bits de contrôle peut paraître élevé par rapport à celui des bits d'information (3 de contrôle pour 4 d'information). Mais il ne s'accroît pas aussi vite que le nombre de bits d'informations ; ainsi, il en faudrait 5 pour contrôler 26 bits.

7. CODES CYCLIQUES REDONDANTS

CODES CYCLIQUES REDONDANTS
CRC

Pour aller vers une détection à 100 % des erreurs dues à la transmission, d'autres codes ont été imaginés. Ils doivent idéalement rester le plus simple possible et exiger le minimum de circuits supplémentaires. Les plus courants sont les *codes de redondance cyclique* (CRC, en abrégé) dont la théorie fait appel à une représentation du message par un polynôme. Rappelons que la formule générale d'un message binaire est un polynôme de la forme :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x^1 + a_0 x^0$$

Les coefficients a_n ne peuvent prendre que des valeurs 0 ou 1 ; en binaire, x vaut 2 et par conséquent, les x^n sont des puissances successives de 2. Ce qui est, d'ailleurs, une définition fondamentale en numération.

Si l'on divise ce polynôme binaire par un autre polynôme, le reste de la division constitue le caractère de contrôle. Commençons par un exemple en décimal.

Soit à transmettre le message 123456 ; si l'on adopte 97 comme polynôme diviseur, on trouvera que :

$$\frac{123456}{97} = 1272 + \frac{72}{97}$$

Le reste, 72, est appelé le CRC. On va le transmettre *avec* le message, ce qui donnera :

$$\underbrace{123456}_{\text{Message}} \quad \underbrace{72}_{\text{CRC}}$$

A la réception, la division du message par le même polynôme, 97, devra donner aussi 72 comme reste. Ce reste sera appliqué avec le CRC transmis à une porte logique OU exclusif, laquelle fournira des zéros partout s'il y a identité. Sinon, c'est qu'une erreur se sera introduite. Ainsi, si l'on recevait 123056, sa division par 97 donnerait 57 comme reste, ce qui est différent de 72.

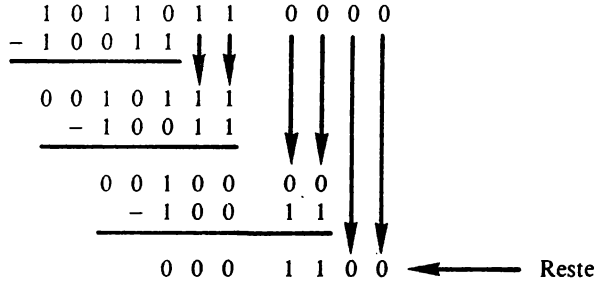
Passons, maintenant, à un exemple en binaire avec, comme diviseur, $x^4 + x + 1$, ce qui donne en binaire 10011. Supposons que le message binaire soit 1011011. Il faut donc diviser 1011011 par 10011 afin de trouver le reste. Cette division est exécutée de la façon suivante :

• On ajoute au dividende — et à sa droite — autant de zéros que le diviseur comporte de bits *moins un*. Ici, donc, on ajoutera à 1011011 quatre zéros, puisque le diviseur compte 5 bits (5 – 1 = 4), ce qui donnera 10110110000. En fait, cela revient à multiplier le dividende par 8.

• Puis, on exécute la division par soustractions successives, en tenant compte qu'il existe deux groupes de bits : le groupe du message, d'origine, et les bits ajoutés (des zéros) à sa droite ; *on ne transmettra aucune retenue* d'un groupe à l'autre.

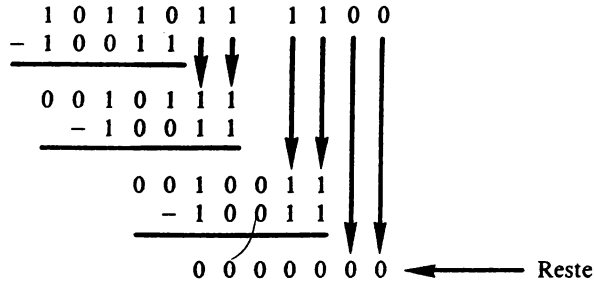
• La dernière ligne de la division donnera le reste.

La division se pose donc ainsi :



Le reste vaut 1100 et c'est le CRC ; c'est lui qu'on va ajouter au message et l'on transmettra réellement :

A la réception, un calcul semblable s'effectue sur le mot reçu, mais il faut, ici, que le reste soit *nul*, faute de quoi c'est qu'une erreur est survenue en cours de route. La vérification s'effectue ainsi :



Le reste étant nul, tout est parfait ! En pratique, la division binaire est assurée par des séries de soustractions et de décalages à droite. On réalise ces opérations à l'aide de portes OU *exclusifs* (soustractions) et de registres à décalages.

Les principaux polynômes diviseurs « normalisés » utilisés pour la création du CRC sont :

Nom	Polynôme
LRCC-8	x^8+1
LRCC-16	$x^{16}+1$
CRC 12	$x^{12}+x^{11}+x^3+x^2+x+1$ ou $(x+1)(x^{11}+x^2+1)$
CRC 16 Forward	$x^{16}+x^{15}+x^2+1$
CRC 16 Backward	$x^{16}+x^{14}+x+1$
CRC-CCITT Forward	$x^{16}+x^{12}+x^5+1$
CRC-CCITT Backward	$x^{16}+x^{11}+x^4+1$

Le CRC-12 est appliqué aux systèmes à mots de 6 bits.

Le CRC-CCITT est le standard européen, pour systèmes à mots de 8 ou 16 bits ; il procure des détections d'erreurs au-delà de 12 bits de message avec une fiabilité de 99 %. Le CRC-16 est un standard américain, assurant 99 % de fiabilité au-delà de 16 bits. Précisons qu'un polynôme est dit *backward*, ou *reverse* (arrière) lorsqu'il est possible de revenir en arrière et d'effectuer un second calcul de CRC, procédure parfois employée avec les bandes magnétiques.

Les schémas pratiques font appel à un registre à décalage à plusieurs sections, séparées par des portes OU exclusifs. L'organisation réelle dépend à la fois de la structure et des normes retenues, la figure 3 en donnant un exemple.

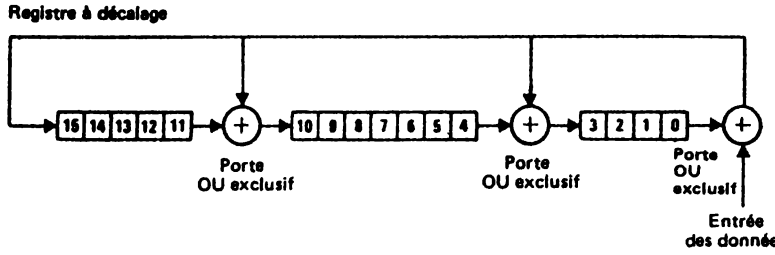


Fig. 3 — Structure du registre à décalage avec ses portes OU exclusifs pour code redondant cyclique.

A l'émission, toutes les cellules du registre sont initialisées à zéro (chargées avec des zéros). Les portes OU exclusifs fournissent un 1 en sortie lorsqu'un 0 et 1 sont simultanément appliqués à leurs deux entrées, et uniquement dans ce cas (deux 0 ou deux 1 à l'entrée donnent 0 en sortie). Si un 1 survient, il en résulte l'état de la figure 4.

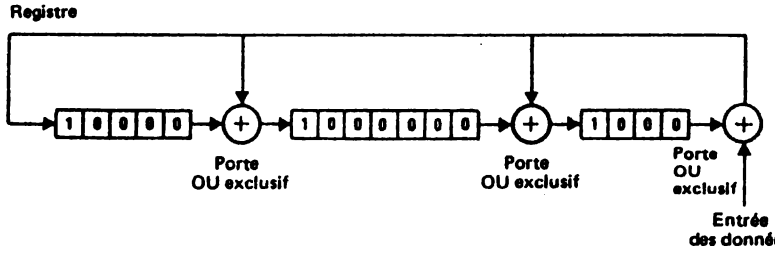


Fig. 4 — Action de l'arrivée d'un 1 lorsque le registre a été, au préalable, initialisé avec des zéros partout.

Le message est par ailleurs transmis et, à la réception, subit le même traitement qu'au poste émetteur. A l'issue du transfert, le registre à décalage de l'émetteur expédie ce mot appliqué lui aussi à l'entrée du registre à décalage récepteur et, *s'il y a identité, le OU exclusif d'entrée ne sort que des zéros* ; les autres OU exclusifs deviennent en quelque sorte « transparents » et laissent passer le bit présent sur leur autre entrée.

Lorsque tous les bits ont défilé, *le registre à décalage du récepteur ne doit plus contenir que des zéros*, cela quel que soit d'ailleurs l'emplacement des portes. L'exception réside avec certaines procédures telles que SDLC ou HDLC où le départ s'effectue non pas avec un chargement de zéros partout mais avec une valeur pré-établie.

CODES POLYNOMIAUX

Le principe, généralisé, montre qu'on utilise un message polynomial de longueur n et un polynôme « générateur » $g(x)$ qui divise $x^n + 1$. L'avis V-41 du *CCITT* préconise l'emploi de codes polynomiaux de longueur $n = 260$, 500, 980 ou 3 860 bits avec le polynôme générateur $g(x) = x^{15} + x^{12} + x^{+1}$ (fig. 5).

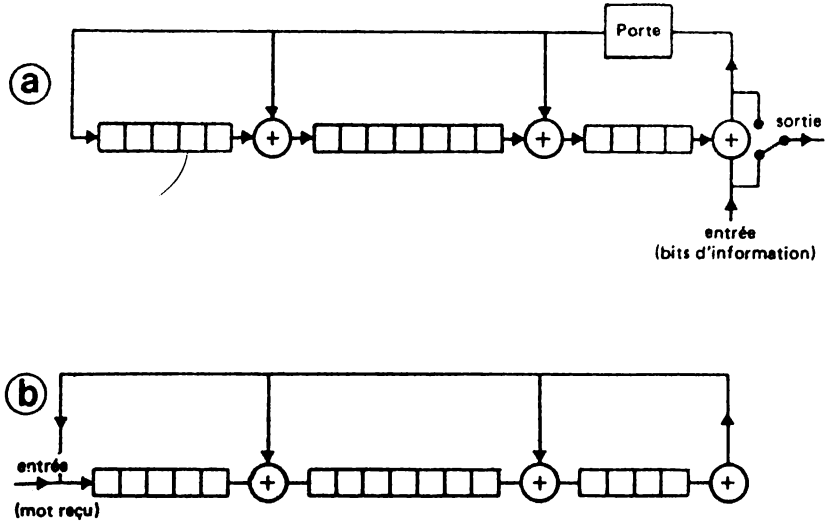
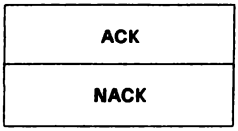


Fig. 5 — Dispositifs de codage *(a)* et de contrôle *(b)* pour codes polynomiaux, selon l'avis V-24 du CCITT.

8. L'ARQ

Quel que soit le code *de détection* d'erreurs utilisé, il convient ensuite de corriger les erreurs détectées. Pour cela, on peut utiliser l'ARQ *« Automatic Request for Repeat »)*, soit *demande de répétition automatique.* Plusieurs procédures sont possibles :



1) Les blocs d'informations sont transmis l'un après l'autre, l'émetteur attendant un accusé de réception après chacun d'eux avant d'envoyer le suivant. Le récepteur envoie un *accusé de réception positif*, ou ACK (de « Acknowledgment ») s'il n'y a pas d'erreur, ou un *accusé de réception négatif* en cas d'erreur, le NACK (« Negative Acknowledgment »). Si l'on espère que la plupart des blocs seront correctement reçus, il y a une perte de temps importante.

2) Les blocs d'informations sont transmis à jet continu, sans l'attente d'un message de réception entre blocs. Ce n'est que lorsque le destinataire détecte une erreur qu'il émet une demande de retransmission. Dans ce cas, et puisque l'émetteur a entamé un bloc suivant lorsque le récepteur a émis son message d'erreur, il lui faut revenir en arrière, ce qui implique qu'il dispose d'une mémoire. Deux stratégies sont à nouveau possibles : *le message est totalement ré-émis à partir du bloc erroné*, ou *seul, le bloc entaché d'erreur est ré-émis.*

On notera que d'autres variantes de la procédure ARQ existent, et ont été utilisées, par exemple par IBM avec le « BSC » ou *« Binary Synchronous Communications ».*

9. CIRCUITS INTÉGRÉS SPÉCIFIQUES

La détection des erreurs, seule, tend à ne plus être suffisante dans les systèmes informatiques et, de plus en plus, on demande que des circuits de correction automatique interviennent. Dans le passé, l'ensemble de ces fonctions faisait appel à des cartes spécialisées, volumineuses et dispendieuses, mais avec le début des années 80, les fabricants de circuits intégrés ont multiplié les circuits spécialisés. Citons, parmi ceux-ci :

- le 2960 de *AMD*, sur 48 broches, qui détecte une erreur en 32 ns max et corrige une erreur simple en 65 ns max ;
- le 8206 de *Intel*, à 68 broches, détectant et corrigeant une erreur en 65 ns max ;
- le DP 8400 de *National Semiconductor*, détectant et corrigeant une erreur simple ou double (sur lequel on va revenir) ;
- les 74 LS 630 et 631, de *Texas*, détectant une erreur en 40 ns max et la corrigeant en 45 ns max (28 broches) ;
- le 74 ALS 632 de *Texas*, 64 broches ;
- le 74 F 418 de *Fairchild*, sur 48 broches (35 ns max pour la détection d'une erreur, 60 ns max pour la correction) ;
- le MC 68540 de *Motorola*, qui semble bien l'un des plus évolués avec le DP 8400 : il détecte et corrige les erreurs simples et détecte les erreurs doubles et les erreurs catastrophiques « tous 1 » ou « tous 0 », mais sans les corriger. Il est en boîtier 48 broches.

CIRCUIT DP 8400

En général, ces circuits font appel à un code de Hamming modifié, en ce sens que les bits de contrôle ne sont pas dispersés dans le mot mais regroupés à sa droite, par exemple. Lors d'une écriture (en mémoire, par exemple), ces circuits calculent le mot de contrôle et l'associent à la donnée ; à la lecture, ils recalculent un mot de contrôle et le comparent au précédent ; s'il y a eu erreur, la correction est aussitôt exécutée par ces circuits. L'étude complète de ces circuits est souvent très complexe, aussi va-t-on évoquer uniquement le DP 8400 de *National Semiconductor*, qui semble bien être l'un des plus évolués puisqu'il détecte jusqu'à trois erreurs et en corrige deux.

Ce circuit est capable d'intervenir dans un cycle mémoire sans que le processeur s'en aperçoive, puisque 60 ns lui suffisent pour exécuter les corrections. Il manipule des mots de 8 ou 16 bits et travaille avec un mot de contrôle sur 6 bits, distinct, selon le code de Hamming modifié.

En résumé :

- il détecte des erreurs simples, ou doubles ;
- corrige ces erreurs, quel que soit leur type (logiciel ou matériel) ;
- détecte 3 erreurs (avec deux DP 8400) ;
- peut travailler sur des mots de 32 bits, ou 48 bits, ou 64 bits... si l'on monte deux, trois ou quatre circuits ;
- il dispose d'*indicateurs* renseignant le système sur le type d'erreurs, leur nombre, leur localisation en mémoire ;
- il dispose de nombreux modes de travail différents ;
- boîtier : 48 broches ;
- détection en 27 ns ;
- correction en 35 ns, typiquement pour une erreur.

Son principe général de fonctionnement est le suivant. Associé au mot de données se trouve un mot de contrôle. Ce mot de contrôle est constitué de *bits de test*. Chacun provient d'un contrôle de parité (OU exclusif) sur un certain nombre de bits du mot de données, opération réalisée par le générateur de *bits de test*. L'ensemble du message (données et *bits de test*) est mis en mémoire.

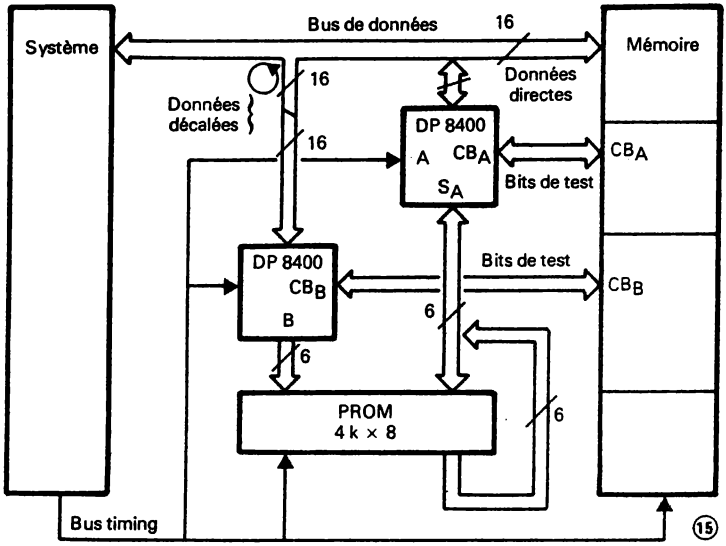


Fig. 6 — Montage de détection et correction de deux erreurs aléatoires, utilisant deux DP 8400.

Lors de la lecture de la mémoire, données et *bits de test* sont entrés dans le DP 8400 qui recalcule les *bits de test* de la donnée lue. Le « générateur de syndrome » compare ces « nouveaux » bits à ceux lus et génère, à son tour, un mot appelé « syndrome ». Les bits de ce syndrome sont, à leur tour, utilisés par le circuit de diagnostic d'erreur (simple, double, données, *bits de test*, etc.) et par le circuit de correction en cas d'erreur simple.

Un montage type du DP 8400 utilisé pour la détection et la correction de deux erreurs aléatoires (logicielles) est donné figure 6. On emploie deux DP 8400 à cet effet.

ORGANISMES COMPÉTENTS

UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

L'*Union Internationale des Télécommunications* (UIT) a été reconnue en 1947 comme l'institution spécialisée de l'ONU dans le domaine des télécommunications. Elle regroupe environ 150 pays membres. Son organe suprême est la « Conférence des plénipotentiaires » ; des « Conférences administratives » mondiales ou régionales sont convoquées pour traiter de questions particulières. La Conférence des plénipotentiaires élit un « Conseil d'Administration » avec un secrétariat général. Les principaux Comités émanant de l'UIT sont les :

• *Comité International d'Enregistrement des Fréquences*, ou IFRB (« International Frequency Registration Board »).

• *Comités Consultatifs Internationaux* (CCI), parmi lesquels :
— le *Comité Consultatif International des Radiocommunications*, ou CCIR ;
— le *Comité Consultatif International Télégraphique et Téléphonique* (CCITT), constitué en 1956, auquel on s'est maintes fois référé dans cet ouvrage à propos des normes.

• L'adresse du CCITT est : 2, rue de Varembé, CH-1211 Genève 20, Suisse.

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Fondée en 1906, la *Commission Electrotechnique Internationale* (CEI) couvre non seulement l'électronique et les télécommunications mais encore l'électricité et l'électrotechnique. La CEI, qui publie des normes, s'est affiliée en 1947 à l'*Organisation Internationale de Normalisation* (ISO).

• L'adresse de la CEI est : 1, rue de Varembé, 1211 Genève 20, Suisse, tél. : (41) 022.34.01.50.

ORGANISATION INTERNATIONALE DE NORMALISATION (ISO)

L'ISO fédère les différentes associations de normalisation des pays du monde ; 65 pays dont leur « Comité membre » représenté et 19 pays qui n'ont pas encore de Comité national de Normalisation étaient membres correspondants de l'ISO en 1979. Le « Comité membre » français est l'AFNOR. Les Comités de quelques autres pays sont :

- DIN : *Deutsches Institut für Norming* pour l'Allemagne
- BSI : *British Standards Institution* pour le Royaume-Uni
- SIS : *Sveriges Standardisk* pour la Suède
- ANSI : *American National Standards Institute* pour les USA
- JISC : *Japonese Industrial Standards Comittee* pour le Japon
- GOST : *Gorodarstvennyj Komitet Standartov* pour l'URSS

L'AFNOR

L'*Association Française de Normalisation*, ou AFNOR, est le « Comité membre » de l'ISO pour la France. Il publie régulièrement des listes complètes de normes, recommandations, projets ou et avant-projets de normes ISO.

- L'adresse de l'AFNOR est : Tour Europe, Cedex 7, 92080 Paris La Défense, tél. (1) 47.78.13.26.

COMITÉ EUROPÉEN DE NORMALISATION (CEN)

Il groupe les représentants des organismes nationaux de normalisation des pays de la « Communauté Européenne » CEE. Le CEN pour objet d'établir des normes européennes (EN), dont le principe est qu'elles entrent sans changement comme normes nationales dans les collections des normes des pays européens dont les comités membres ont voté en faveur de leur adoption.

L'ECMA

L'ECMA, *European Computer Manufacturers Association*, est constituée par les constructeurs de matériels informatiques en Europe et publie également des normes d'une grande importance professionnelle, bien que sans statut officiel.

- L'adresse d'ECMA est : 114, rue du Rhône, CH-1204 Genève.

COMITÉ EUROPÉEN DE NORMALISATION ÉLECTROTECHNIQUE (CENELEC)

Le CENELEC a été créé en 1972. Les normes européennes (EN) établies par le CENELEC sont intégrées dans les collections propres des pays européens selon le même principe de mise en application nationale des normes du CEN.

QUELQUES ADRESSES EN FRANCE

• La *Direction Générale des Télécommunications* (DGT) dispose d'un Service de la Téléinformatique au 20, avenue de Ségur, 75007 Paris, tél. 43.27.35.81.
• L'*Agence Commerciale Téléinformatique et Réseaux Spécialisés* est installée au 32, bd de Vaugirard, 75731 Paris Cedex 15.
• *Transpac* a pour adresse : Tour Maine-Montparnasse, BP 145, 75755 Paris Cedex 15.
• Le *Centre de Promotion de la Téléinformatique* est installé au 8-10, bd de Vaugirard, 75738 Paris Cedex 15.
• Le *Centre National des Télécommunications* (CNET) a pour adresse : 38-40, rue du Général Leclerc, 92131 Issy-les-Moulineaux.
• Le *Centre Commun d'Etudes de Télévision et de Télécommunications* (CCETT), émanation de la DGT et de TéléDiffusion de France (TDF) traite de : 1) la transmission des images (télévision, visiophone) et des documents (télécopie) par les nouveaux procédés ; 2) la télédistribution ; 3) les nouveaux services audiovisuels sur téléviseurs ; 4) les réseaux de télé-informatique.
• *TéléDiffusion de France* (TDF), pour les services tels qu'Antiope : Délégation à l'information, 21-27, rue Barbès, 92120 Montrouge.

ORGANISMES INTERNATIONAUX

• *ANSI (American National Starndards Institute)*, 1430 Broadway, New York, NY 10018, USA.
• *DIN (Deustches Institute für Normung)*, Burggrafenstrasse 4-7, D-1000 Berlin 30, Allemagne.
• *ASME (American Society of Mechanical Engineers)*, 345 East, 47th Street, New York, NY 10017, USA.
• *EIA (Electronic Industries Association)*, 2001 I Street, N.W. Washington, DC 20006, USA.
• *IEEE (Institute of Electrical and Electronic Engineers)*, 345 East 47th Street, New York, NY 10017, USA.
• *ISA (Instrument Society of America)*, 67 Alexander Drive, POB 12277, Research Triangle Park, NC 27009, USA.
• *NBS (National Bureau of Standards)*, Washington, DC 20234, USA.

MOTS-CLÉS

A

• *A/D* : conversion analogique-digitale, 216
• *ADC* : convertisseur analogique numérique, 216
• *ACIA* : un circuit UART pour liaison asynchrone, 56
• *ACK* : accusé de réception positif, 274
• *ACK* : protocole de dialogue pour imprimante, 85
• *Acknowledge* : signal d'échange pour imprimante, 22
• *A/N* : conversion analogique-numérique, 216
• *Approximations successives* : mode de conversion A/N, 227
• *ARQ* : mode de détection des erreurs, 258, 274
• *ASCII* : code, 45
• *Asynchrone* : mode, 38

B

• *Bande passante* : des moniteurs, 171
• *Bandes magnétiques* : pour micro-informatique, 180
• *Baud* : unité de débit, 262
• *Baudot* : code, 42
• *Bipolaire* : code binaire, 242
• *Biphases* : codes binaires, 246
• *Bits par seconde* : unité de débit, 254
• *Boucle 20 mA* : liaison asynchrone série, 52
• *Boucle 60 mA* : liaison asynchrone série, 53
• *Bus de systèmes à micro-processeur*, 11
• *Bus d'extension de base* : pour micro-ordinateurs, 14
• *Bus d'extension de l'Apple IIe* : caractéristiques, 138
• *Bus d'extension de l'IBM-AT* : caractéristiques, 142
• *Bus d'extension de l'IBM-PC* : caractéristiques, 134
• *Bus d'extension des systèmes*, 112
• *Bus du microprocesseur*, 10
• *Bus LSI-11* : caractéristiques, 140
• *Bus pour disquettes*, 153
• *Busy* : signal d'échange pour imprimante, 22
• *BUSY* : protocole de dialogue pour imprimante, 83

C

• *Câble pour imprimante parallèle*, 24
• *CAD* : convertisseur analogique-digital, 216
• *CAN* : convertisseur analogique-numérique, 216
• *CDA* : convertisseur digital-analogique, 216
• *Caractère de synchronisation* : en mode synchrone, 97
• *Cartouche* : pour bande magnétique, 180
• *Cassette* : pour bande magnétique, 180
• *Centronics* : norme pour commande d'imprimante parallèle, 21
• *Circuits intégrés pour disquettes*, 154
• *Circuit ADC 80* : convertisseur intégré A/N, 237
• *Circuit DP 8400* : de détection et correction d'erreurs, 275/9

• *Circuit WD 279X* : CI pour disques, 154
• *Circuit 6852* : interface synchrone, 99
• *Circuit 8088* : brochage du microprocesseur, 9
• *Circuit 8251* : un USART, 106
• *Circuit 8255* : interface parallèle, 19
• *Circuit 8273* : contrôleur SDLC, 109
• *Circuits 8291 et 8292* : circuits intégrés IEEE-488, 208
• *Clavier* : commande par interface parallèle, 30
• *Clavier de l'IBM-PC* : commandes et liaisons, 32
• *Code télex CCITT n° 2*, 43
• *Codes CCITT*, 43
• *Codes cycliques redondants* : pour détecter les erreurs, 269
• *Codes 2/5* : pour détecter les erreurs, 263
• *Commande de téléimprimeur par boucle 20 mA*, 54
• *Comptage* : mode de conversion A/N, 225
• *CNA* : convertisseur numérique-analogique, 216
• *Connecteurs pour imprimantes série* : en RS-232C, 79
• *Connecteur 25 broches d'imprimante parallèle (IBM-PC)*, 25
• *Connecteur 36 broches d'imprimante parallèle (IBM-PC)* : 26
• *Conversion parallèle-série*, 49
• *Conversion série-parallèle*, 50
• *CRC* : code cyclique redondant, pour détecter les erreurs, 269
• *Coupleur optique* : pour liaisons série, 55

D

• *D/A* : conversion digital-analogique, 216
• *DAC* : convertisseur digital-analogique, 216
• *DCE* : « data circuit termination », 62
• *Détection du start* : en mode asynchrone, 50
• *Disquette IBM-PC* : caractéristiques, 156
• *DM* : ou code Miller, 248
• *Double complémentation* : pour situer une erreur, 265
• *DTE* : « data terminal equipment », 62

E

• *Echelle R/2R* : mode de conversion N/A, 234
• *Echo* : mode de détection d'erreurs, 257
• *Emission doublée* : pour détecter les erreurs, 257
• *Erreurs* : détection et correction, 256
• *ESDI* : norme pour disques durs, 148 et 161

F

• *Faux départ* : en mode asynchrone série, 50
• *FEC* : mode de détection et correction des erreurs, 258
• *Feux de croisement* : commande industrielle en mode parallèle, 35
• *Flash* : mode de conversion A/N, 230
• *FM* : code à modulation de fréquence, 249
• *Formats pour disquettes IBM*, 160
• *FSK* : modulation en fréquence pour norme Kansas-City, 187
• *Future-Bus IEEE-P896* : un bus normalisé, 127

G

• *GCR* : code pour enregistrement magnétique, 252
• *GP-IB* : autre nom du bus IEEE-488, 196
• *Graphisme-couleur sur IBM-PC* : description, 174

H

• *Hamming* : codes de détection d'erreurs et de correction, 266
• *Haut-parleur de l'IBM-PC* : son connecteur, 178
• *HP-IB* : autre nom du bus IEEE-488, 196
• *HP-IL* : bus série de Hewlett-Packard, 209

I

• *IEC 625-1* : norme CEI correspondant au IEEE-488, 196
• *IEEE-P896* : normalisation du Future-Bus, 127
• *IEEE-488* : bus normalisé, 196
• *Imparité* : pour corrections d'erreurs, 46
• *Imparité* : d'un mot binaire, 258
• *Imprimante en commande parallèle*, 21
• *Imprimante en mode série*, 77
• *Imprimante série avec IBM-PC*, 89
• *Interface disquettes IBM-PC*, 160
• *Interface cassette sur IBM-PC*, 194
• *Interface synchrone sur IBM-PC* : avec le SDLC, 108
• *Isochrone* : l'un des modes synchrone, 98

K

• *Kansas-City* : norme pour cassettes magnétiques, 185

L

• *Liaison parallèle* : les principes, 18
• *Liaison série sur IBM-PC*, 85
• *Liaisons synchrones* : avantages et inconvénients, 96

M

• *Macintosh* : liaison série, 91
• *Macintosh et imprimantes*, 93
• *Macintosh et disques* : la liaison, 166
• *Manettes de jeux pour l'IBM-PC*, 213
• *MFM* : code FM modifié, 249
• *Microdisquette 3"1/2* : caractéristiques, 150
• *Microprocesseur* : son organisation interne, 8
• *Miller* : code binaire, 248
• *MMFM* : code MFM modifié, 251
• *Mode message* : pour liaison synchrone, 97
• *Monostable* : mode conversion A/N, 238
• *Multibus* : un bus normalisé, 117
• *Multibus II* : un bus normalisé, 127
• *Multiniveau* : codes binaires, 253
• *M2FM* : autre nom du code MMFM, 251

N

• *N/A* : conversion numérique-analogique, 216
• *NACK* : accusé de réception négatif, 274
• *Normes TV*, 170
• *NRZ* : codes sans retour à zéro, 242
• *NRZI* : code NRZ inversé, 244

P

• *Parallèle* : mode de conversion A/N, 230
• *Parité* : pour correction d'erreurs, 46
• *Parité* : d'un mot binaire, 258
• *Parité longitudinale*, 261
• *Parité transversale*, 259
• *Parité verticale*, 259
• *Péritel* : prise pour péri-télévision, 176
• *PIA* : interface parallèle, 19
• *Polaire* : code binaire, 242
• *Polynomiaux* : codes de détection d'erreurs, 273
• *Protocoles pour imprimantes* : le dialogue, 82

Q

• *QIC-02* : norme d'interfaçage pour bande quart de pouce, 182

R

• *Rampe* : mode de conversion A/N, 221
• *Résistances pondérées* : mode de conversion N/A, 232
• *Retransmission* : pour détecter les erreurs, 257
• *RS-232* : norme de liaison série, 64
• *RS-232C* : plus récente version de la norme 232, 64
• *RS-422* : norme de liaison série, 72
• *RS-423* : norme de liaison série, 72
• *RZ* : code à retour à zéro, 245

S

• *SASI* : norme pour disques, 148
• *SA 450* : norme pour disquettes, 148
• *SCSI* : norme pour disques, 148
• *Signaux pour moniteurs* : principes, 170
• *Start-stop* : synonyme de mode asynchrone, 41
• *Start-stop* : mode pour bande magnétique, 182
• *STD* : un bus normalisé, 115
• *Streamer* : mode pour bande magnétique, 182
• *Strobe* : signal d'échange pour imprimante, 22
• *ST 506* : norme pour disques, 148 et 161
• *Synchrone* : mode, 38
• *S-100* : un bus normalisé, 120

T

• *Ternaire* : code ternaire, 242

U

• *UART* : circuit de liaison asynchrone, 56
• *UART à téléimprimeur* : liaison série, 74
• *Unipolaire* : code binaire, 242
• *Unité à disquettes 3"1/2* : caractéristiques, 152
• *Unité à disquettes 5"1/4* : caractéristiques, 151
• *USART* : récepteur-émetteur universel synchrone-asynchrone, 104

V

• *Versabus* : un bus normalisé, 119
• *Vidéo composite* : ce que c'est, 170
• *Vidéo monochrome sur IBM-PC* : description, 171
• *VME* : un bus normalisé, 123
• *V.10* : norme de liaison série, 72
• *V.11* : norme de liaison série, 72
• *V.24* : norme de liaison série semblable à la RS-232C, 64

X

• *XON/XOFF* : protocole de dialogue pour l'imprimante, 83
• *XT d'IBM* : liaison au disque fixe, 165

Z

• *Z-Bus* : un bus normalisé, 119

TABLE DES MATIÈRES

Avant-propos 5

Chapitre I — **Autour du microprocesseur** 7
1. Organisation de base d'un système 8
2. Les diverses organisations internes 11
3. Le bus d'extension de base 14

Chapitre II — **Liaisons en parallèle** 17
1. Principes 18
2. Commande d'une imprimante : la liaison Centronics 21
3. Le câble de liaison à l'imprimante 24
4. Commande de clavier 30
5. Commande du clavier de l'IBM-PC 32
6. Applications industrielles 33

Chapitre III — **Liaisons série asynchrones : principes** 37
1. Modes synchrone et asynchrone 38
2. Origine électromécanique 39
3. Transposition électronique : le mode start-stop 41
4. Les codes 42
5. Protection contre les erreurs 46
6. Circuits d'émission 49
7. Principe des circuits de réception 50
8. Les boucles 20 et 60 mA 51
9. Liaison par coupleur optique 55
10. Liaisons en V.24 et RS-232 56
11. UART et USART 56
12. Une interface asynchrone type : l'ACIA 56

Chapitre IV — Liaisons série : normes RS-232 et V.24 61
1. Les DTE et DCE 62
2. Les RS-232 et V.24 67
3. Normes RS-423 et 422 (V.11 et V.10) 72
4. Quelques montages types 73
5. Principe des échanges avec une imprimante 77
6. Connecteurs pour imprimantes série 78
7. Protocoles de dialogue avec l'imprimante 82
7.1. Mode Busy par caractère 83
7.2. Mode Busy par bloc 83
7.3. Mode XON/XOFF 83
7.4. Mode ACK 85
8. Liaison série sur l'IBM-PC 85
9. Usage d'une imprimante série avec l'IBM-PC 89
10. Les liaisons série du Macintosh 91

Chapitre V — Communications synchrones série 95
1. Avantages et inconvénients 96
2. Le signal de synchronisation 97
3. Le mode isochrone 98
4. Un circuit d'interface synchrone type 99
5. L'USART 104
6. L'interface synchrone de l'IBM-PC 108

Chapitre VI — Bus systèmes issus des microprocesseurs 111
12. Les bus d'extension 112
2. Bus STD 115
3. Le Multibus 117
4. Le Z-Bus 119
5. Le Versabus 119
6. Le bus S-100 120
7. Le bus VME 123
8. Le Future Bus IEEE-P896 127
9. Le Multibus II 127

Chapitre VII — Bus d'extension des systèmes micro-ordinateurs 133
1. Les bus d'extension de l'IBM-PC 134
2. Bus d'extension de l'Apple II*e* 138
3. Bus du LSI-11 140
4. Bus d'extension de l'IBM-AT 142

Chapitre VIII — Liaison avec les disques 147
1. Interfaces pour disques et disquettes 148
2. Bus pour unités à disquettes 150
3. Circuits intégrés d'interface 154
4. Unités à disquettes de l'IBM-PC 156
5. Interfaces pour disques durs ST 506 et ESDI 161
6. Liaison au disque fixe de l'IBM PC-XT 165
7. La liaison disque du Macintosh 166

Chapitre IX — Interfaces avec les moniteurs 169
1. Les principes de la commande d'un moniteur 170
2. La sortie vidéo monochrone sur l'IBM-PC 171
3. La sortie graphique-couleur sur l'IBM-PC 174
4. La prise Péritel 176
5. L'interface HP de l'IBM-PC 178

Chapitre X — Interfaces pour bandes magnétiques en cassettes ou cartouches 179
1. Cartouches à bande quart de pouce 180
2. Modes start-stop et streamer 182
3. Utilisation de cassettes 184
4. Montages aux normes Kansas-City 187
5. Interface cassette de l'IBM-PC 194

Chapitre XI — Les bus HP-IB (IEEE-488) et HP-IL 195
1. HP-IB et IEEE-488 196
2. Caractéristiques et structures 196
3. Description mécanique 200
4. Adressage et dialogue 205
5. Circuits intégrés spécialisés IEEE-488 206
6. Le bus HP-IL 209
6.1. Transmission des messages 210
6.2. Exemple d'application avec un multimètre 212

Chapitre XII — Interfaçage avec le monde analogique 213
1. Introduction 214
2. Conversion analogique-numérique 216
2.1. Rampe 219
2.2. Comptage 223
2.3. Approximations successives 225
2.4. Flash, ou parallèle 228

3. Convertisseurs numérique-analogique 228
3.1. Résistances pondérées 230
3.2. Réseau R/2R 232
4. Exemple de montage A/D 235
5. Commande de manettes de jeux de l'IBM-PC 236

Chapitre XIII — Codes binaires 239
1. Codes unipolaire, polaire, bipolaire 240
2. Codes NRZ 240
2.1. Le NRZI 242
3. Codes RZ 243
4. Codes biphases L, M et S 244
5. Codes Miller (DM) 246
6. Codes FM et MFM 247
7. Le code GCR 250
8. Codes multiniveaux 251
9. Bits par seconde et bauds 252

Chapitre XIV — Détection et correction d'erreurs 255
1. Les erreurs et leur traitement 256
2. Parité transversale 258
3. Parité longitudinale 261
4. Utilisation des codes 2/5 263
5. Correction par double complémentation 265
6. Codes de Hamming 266
7. Codes cycliques redondants 269
8. L'ARQ 274
9. Circuits intégrés spécifiques 274

Organismes compétents 277

Mots-clés 281

EDITIONS RADIO

Service lecteurs

(à retourner à S.E.C.F.-Éditions Radio, 9, rue Jacob, 75006 Paris)

Pour nous permettre de vous proposer des ouvrages toujours meilleurs, nous souhaiterions recevoir vos critiques, appréciations et suggestions sur le présent livre :

_____

_____

_____

_____

Quels sont les ouvrages (thème, sujet, niveau) que vous souhaiteriez voir publier par notre société ?

_____

_____

_____

_____

Nous vous remercions de votre confiance et de votre coopération.

S.E.C.F.-Éditions Radio

Je désire recevoir gratuitement et sans engagement (mettre une croix dans la case) :

☐ Votre catalogue général (Electronique professionnelle et grand public, Informatique, Hi-Fi, Vidéo)

Nom : _____ Prénom : _____

Adresse : _____

_____

Secteur d'activité et fonction : _____

_____

CENTRES D'INTÉRÊTS

☐ Electronique professionnelle
☐ Electronique de loisirs
☐ Vidéo
☐ Hifi, CB...
☐ Micro-informatique professionnelle
☐ Micro-informatique de loisirs
☐ Autres : .....

_____

Voir au dos "Correspondance Auteurs"

TECHNIQUE-NORMES-APPLICATIONS

DES

INTERFACES

POUR MICRO-ORDINATEURS

Si les microprocesseurs ont déjà fait l'objet de très nombreux livres, leurs interfaces avec les périphériques et le monde extérieur ont généralement été négligées. Ce livre vient donc à point et comble une lacune grave. Il développe sur l'angle à la fois théorique et pratique des thèmes essentiels tels que : les liaisons en mode parallèle, en mode série (synchrone et asynchrone), les normes (RS-232, V.24, etc.), les protocoles de liaisons, la réalisation des câbles de liaisons (imprimantes, par exemple), les bus avec leurs normes et leurs caractéristiques (bus d'extension, Multibus, VME, etc.), les liaisons avec les disques (normes ST 506, ESDI...), les interfaces avec les moniteurs, claviers, unités à bandes magnétiques, les liaisons IEEE-488 (HP-IB), l'interfaçage avec le monde industriel et les conversions analogique-numérique et vice-versa, les codes binaires, les principes et les applications de la détection et de la correction des erreurs, etc. Le tout est largement illustré, en particulier au travers des applications à l'IBM-PC.

9 782709 109864
ISBN 2 7091 0986 7
Code 120

EDITIONS RADIO

F 130/36/6